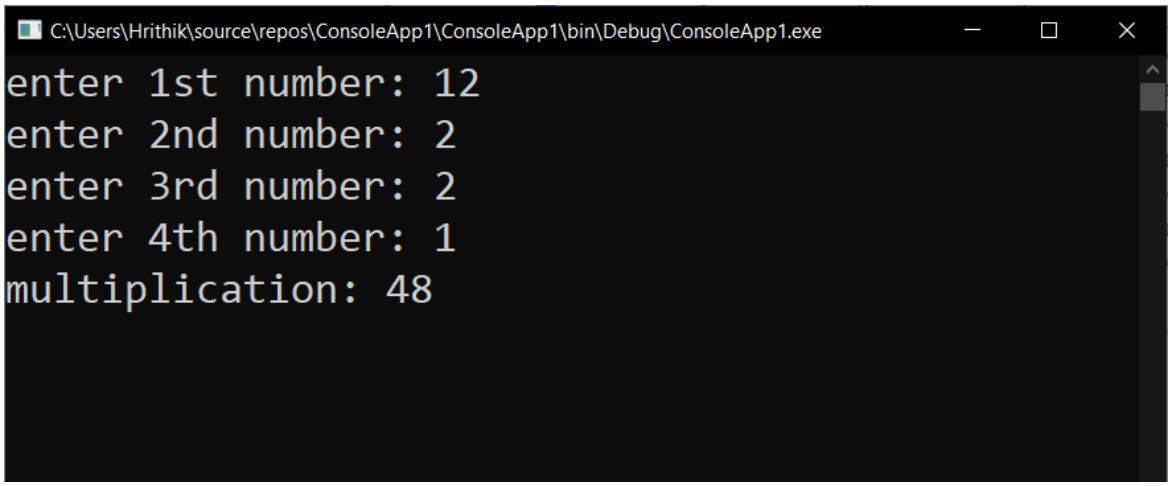
1. Working with basic C# and ASP .NET

1a Aim: Create an application that obtains four int values from the user and displays the product.

Source Code:

```
using System;

namespace ConsoleApp1 {
    class Program {
        static void Main(string[] args) {
            Console.Write("enter 1st number: ");
            int num1 = Int32.Parse(Console.ReadLine());
            Console.Write("enter 2nd number: ");
            int num2 = Int32.Parse(Console.ReadLine());
            Console.Write("enter 3rd number: ");
            int num3 = Int32.Parse(Console.ReadLine());
            Console.Write("enter 4th number: ");
            int num4 = Int32.Parse(Console.ReadLine());

            Cons2ole.WriteLine("multiplication: " + num1 * num2
* num3 * num4);
            Console.ReadKey();
        }
    }
}
```

Output:

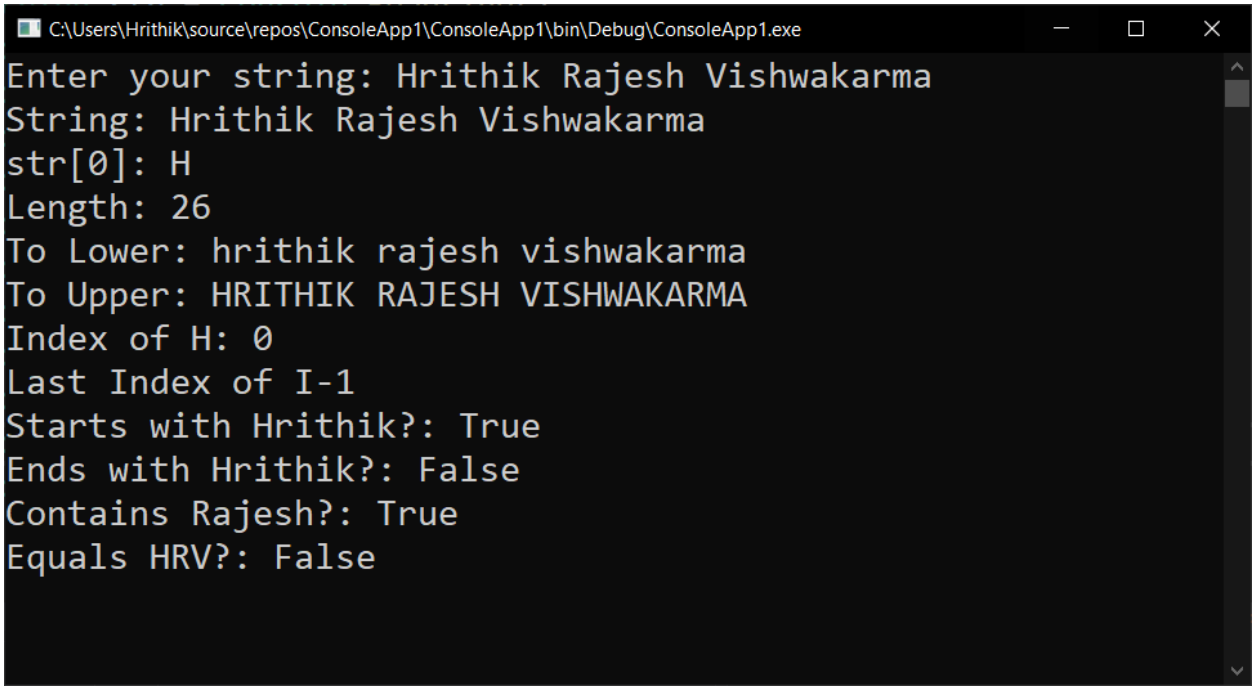


1b Aim: Create an application to demonstrate string operations.

Source Code:

```
using System;

namespace ConsoleApp1 {

    class Program {

        static void Main(string[] args) {

            Console.Write("Enter your string: ");
            String str = Console.ReadLine();
            Console.WriteLine("String: " + str);

            Console.WriteLine("str[0]: "+ str[0]);
            Console.WriteLine("Length: " + str.Length);
            Console.WriteLine("To Lower: " + str.ToLower());
            Console.WriteLine("To Upper: " + str.ToUpper());
            Console.WriteLine("Index of H: " +
str.IndexOf("H"));
            Console.WriteLine("Last Index of I" +
str.LastIndexOf("I"));
            Console.WriteLine("Starts with Hrithik?: " +
str.StartsWith("Hrithik"));
            Console.WriteLine("Ends with Hrithik?: " +
str.EndsWith("Hrithik"));
            Console.WriteLine("Contains Rajesh?: " +
str.Contains("Rajesh"));
            Console.WriteLine("Equals HRV?: " +
str.Equals("HRV"));

            Console.ReadKey();

        }
    }
}
```

Output:



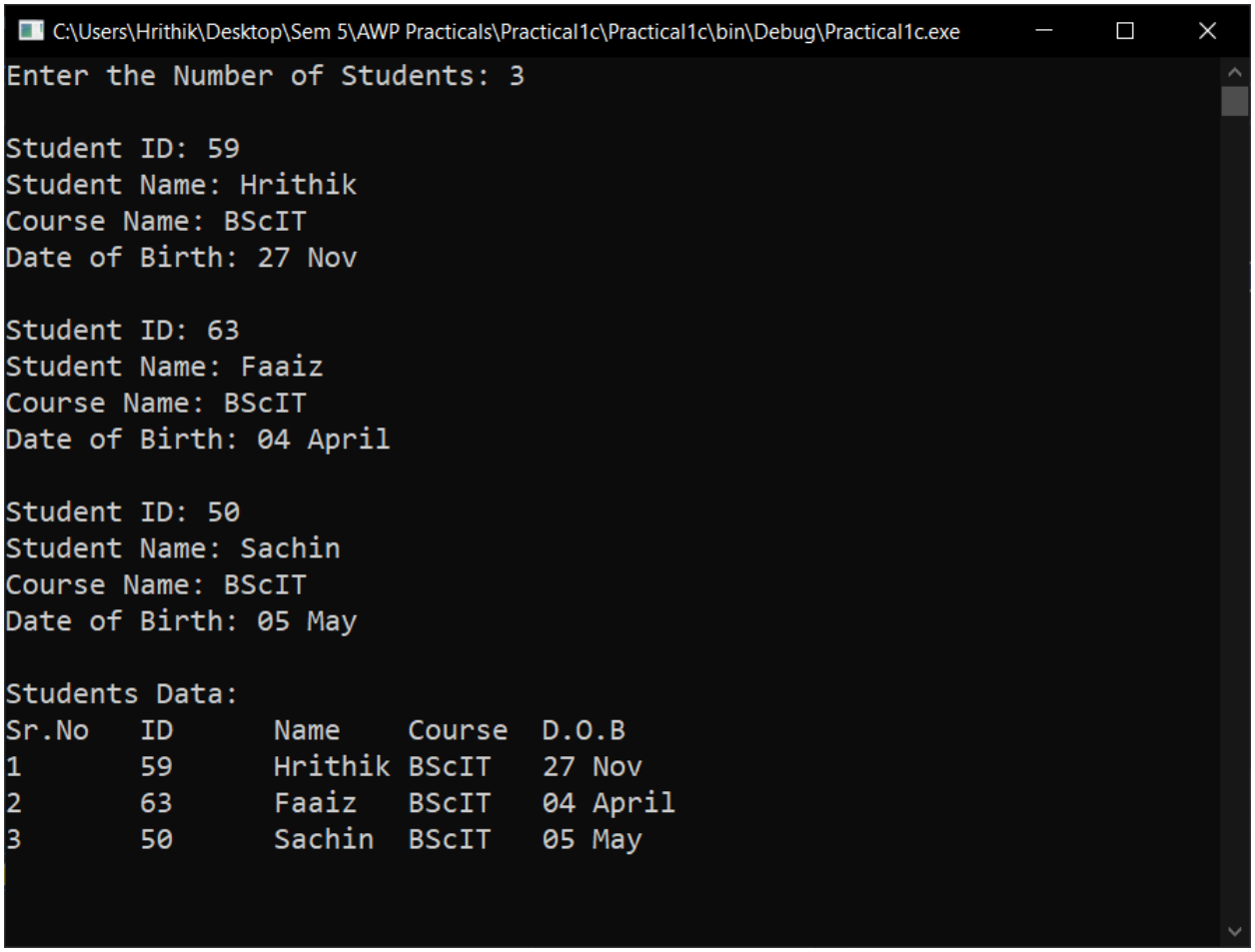
1c Aim: Create an application that receives the (Student Id, Student Name, Course Name, Date of Birth) information from a set of students. The application should also display the information of all the students once the data entered.

Source Code:

```
using System;

namespace Practical1c {
    class Program {
        static void Main(string[] args) {
            Console.Write("Enter the Number of Students: ");
            int n = Int32.Parse(Console.ReadLine());
            Student student = new Student(n);
            student.takeInput();
            student.printData();
            Console.ReadKey();
        }
    }

    public class Student {
        public int n;
        public Student[] students;
```

```
        public String ID, Name, Course, DOB;

        public Student(int n) {
            this.n = n;
            students = new Student[n];
        }
        public Student(String id, String name, String course,
String dob) {
            this.ID = id;
            this.Name = name;
            this.Course = course;
            this.DOB = dob;
        }
        public void takeInput() {
            for(int i = 0; i < n; i++)
            {
                Console.WriteLine();
                Console.Write("Student ID: ");
                String ID = Console.ReadLine();
                Console.Write("Student Name: ");
                String Name = Console.ReadLine();
                Console.Write("Course Name: ");
                String Course = Console.ReadLine();
                Console.Write("Date of Birth: ");
                String DOB = Console.ReadLine();

                students[i] = new Student(ID,Name,Course,DOB);
            }
        }
        public void printData() {
            Console.WriteLine("\nStudents Data: ");
            Console.WriteLine("Sr.No\tID\tName\tCourse\tD.O.B");
            for (int i = 0; i < n; i++) {
                Console.Write((int)(i + 1)
+"\t"+students[i].ID+"\t"+ students[i].Name + "\t" +
students[i].Course + "\t" + students[i].DOB + "\t" + "\n");
            }
        }
    }
}
```

Output:



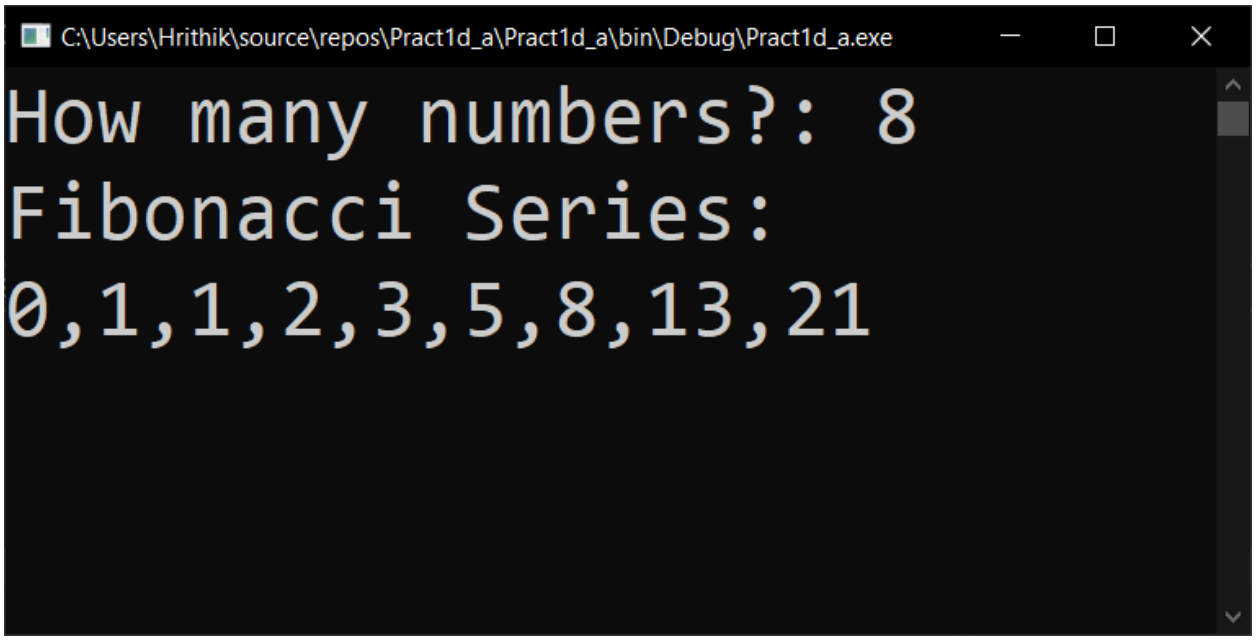
1d Aim: Create an application to demonstrate following operations.

a) Source Code (Generate Fibonacci series):

```
using System;

namespace Pract1d_a {
    class Program {
        static void Main(string[] args) {
            Console.Write("How many numbers?: ");
            int num = Int32.Parse(Console.ReadLine());
            Console.Write(fibo(num));

            Console.ReadKey();
        }
```

```
        static String fibo(int num) {
            int[] fibArr = new int[num+1];
            fibArr[0] = 0;
            fibArr[1] = 1;
            for(int i = 2; i <= num; i++) {
                fibArr[i] = fibArr[i - 1] + fibArr[i - 2];
            }

            return "Fibonacci Series: \n" + string.Join("," ,
fibArr);
        }
    }
}
```

Output:

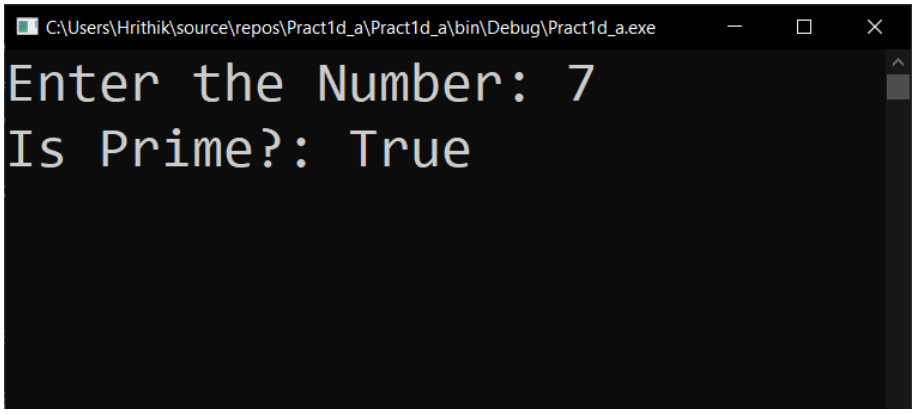


b) Source Code (Test for prime numbers):

```
using System;

namespace Pract1d_a {
    class Program {
        static void Main(string[] args) {
```

```
            Console.Write("Enter the Number: ");
            int num = Int32.Parse(Console.ReadLine());
            Console.Write("Is Prime?: " + isPrime(num));

            Console.ReadKey();
        }

        static Boolean isPrime(int num) {
            for(int i = 2; i < Math.Sqrt(num); i++) {
                if(num%i == 0) return false;
            }
            return true;
        }
    }
}
```

Output:

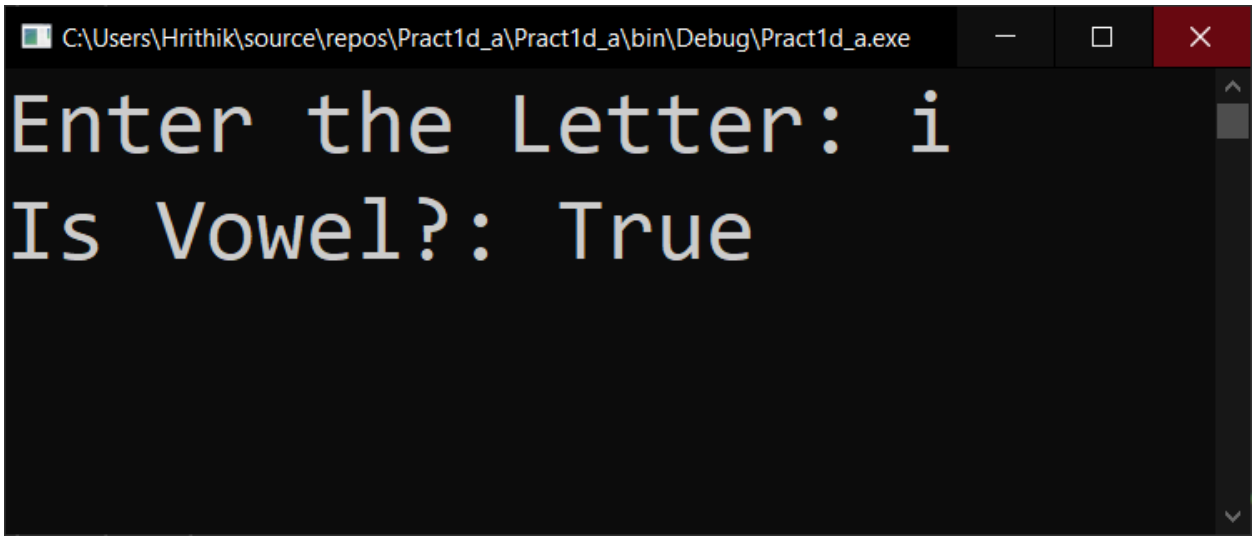


c) Source Code (Test for vowels):

```
using System;

namespace Pract1d_a {
    class Program {
        static void Main(string[] args) {
            Console.Write("Enter the Letter: ");
            String ltr = Console.ReadLine();
            Console.Write("Is Vowel?: " + isVowel(ltr));

            Console.ReadKey();
```

```
        }

        static Boolean isVowel(String ltr) {
            String vowel = "aeiou";
            if(vowel.Contains(ltr.ToLower()) return true;
            return false;
        }
    }
}
```

Output:



d) Source Code (Use of foreach loop with arrays):

```
using System;

namespace Pract1d_a {
    class Program {
        static void Main(string[] args) {
            String[] dummyArr = { "Hrithik", "Ryan", "Margot",
"Patrick" };
            Console.WriteLine("Names: ");

            foreach(String name in dummyArr) {
                Console.Write(name + " ");
            }

            Console.ReadKey();
```

```
        }
    }
}
```

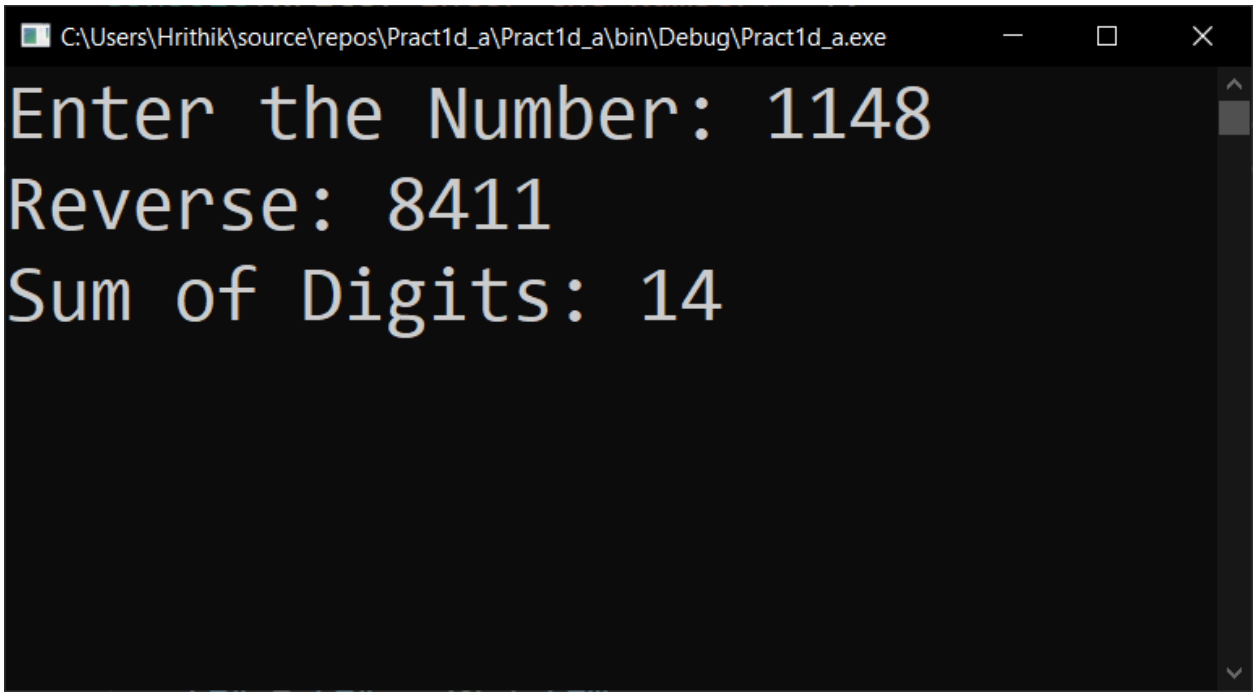
Output:

e) Source Code (Reverse a number and find sum of digits of a number):

```
using System;

namespace Pract1d_a {
    class Program {
        static void Main(string[] args) {
            Console.Write("Enter the Number: ");
            int num = Int32.Parse(Console.ReadLine());
            Console.WriteLine("Reverse: " + reverseNum(num));
            Console.Write("Sum of Digits: " + sumOfDigits(num));

            Console.ReadKey();
        }
        static int reverseNum(int num) {
            int rev = 0;
            while(num != 0) {
                int rem = num % 10;
                rev = rev * 10 + rem;
                num = num / 10;
            }
            return rev;
        }
```

```
        static int sumOfDigits(int num) {
            int sum = 0;
            while (num != 0) {
                int rem = num % 10;
                sum = sum + rem;
                num = num / 10;
            }
            return sum;
        }
    }
}
```

Output:



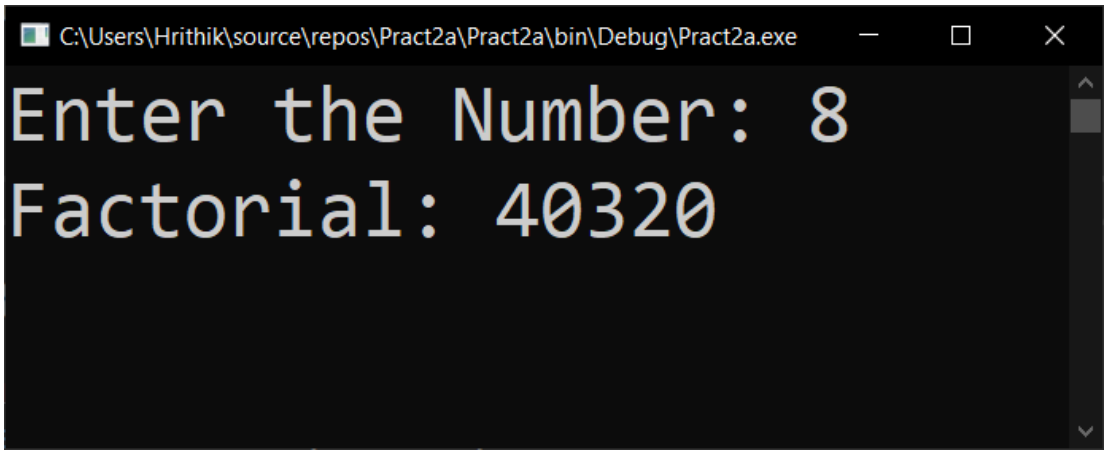
2. Working with Object Oriented C# and ASP .NET

2a Aim: Create simple application to perform following operations.

a) Source Code (Finding factorial Value):

```
using System;

namespace Pract2a {
    class Program {
        static void Main(string[] args) {
            Console.Write("Enter the Number: ");
            int num = Int32.Parse(Console.ReadLine());
            Console.WriteLine("Factorial: " + fact(num));
            Console.ReadKey();
        }

        static int fact(int num) {
            int fact = 1;
            for(int i = 1; i <= num; i++) {
                fact *= i;
            }
            return fact;
        }
    }
}
```

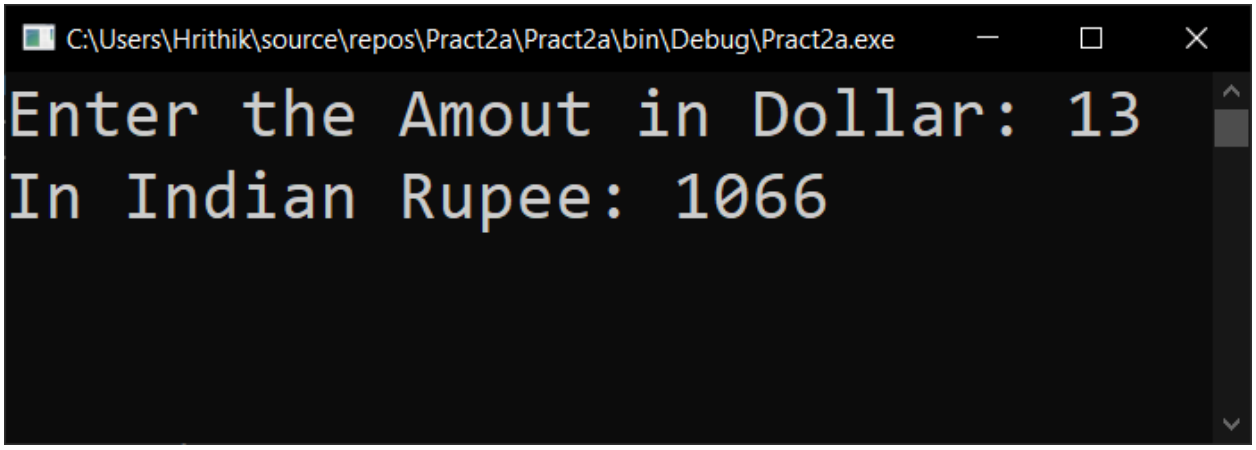
Output:



b) Source Code (Money Conversion):

```
using System;

namespace Pract2a {
    class Program {
        static void Main(string[] args) {
            Console.Write("Enter the Amout in Dollar: ");
            int dollar = Int32.Parse(Console.ReadLine());
            Console.WriteLine("In Indian Rupee: " +
dollarToInr(dollar));
            Console.ReadKey();
        }

        static int dollarToInr(int dollar) {
            int dollarToInr = 82;
            return dollar * dollarToInr;
        }
    }
}
```

Output:

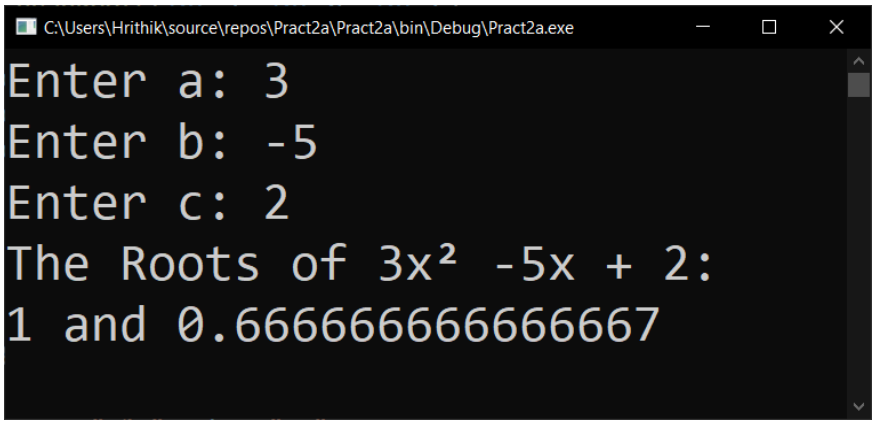


b) Source Code (Quadratic Equation):

```
using System;

namespace Pract2a {
    class Program {
```

```
        static void Main(string[] args) {
            Console.Write("Enter a: ");
            int a = Int32.Parse(Console.ReadLine());
            Console.Write("Enter b: ");
            int b = Int32.Parse(Console.ReadLine());
            Console.Write("Enter c: ");
            int c = Int32.Parse(Console.ReadLine());
            Console.WriteLine("The Roots of " + formatQuadEq(a,
b , c) + ": \n" + quadRoots(a, b, c));
            Console.ReadKey();
        }

        static String quadRoots(int a, int b, int c) {
            double roota = (-b + (Math.Sqrt((b * b) - (4 * a *
c)))) / (2 * a);
            double rootb = (-b - (Math.Sqrt((b * b) - (4 * a *
c)))) / (2 * a);
            return roota.ToString() + " and " +
rootb.ToString();
        }

        static string formatQuadEq(int a, int b, int c) {
            if(b < 0 && c < 0) return a + "x² " +b+ "x " + c;
            if(b < 0 && c > 0) return a + "x² " +b+ "x + " + c;
            return a + "x² + " + b + "x + " + c;
        }
    }
}
```

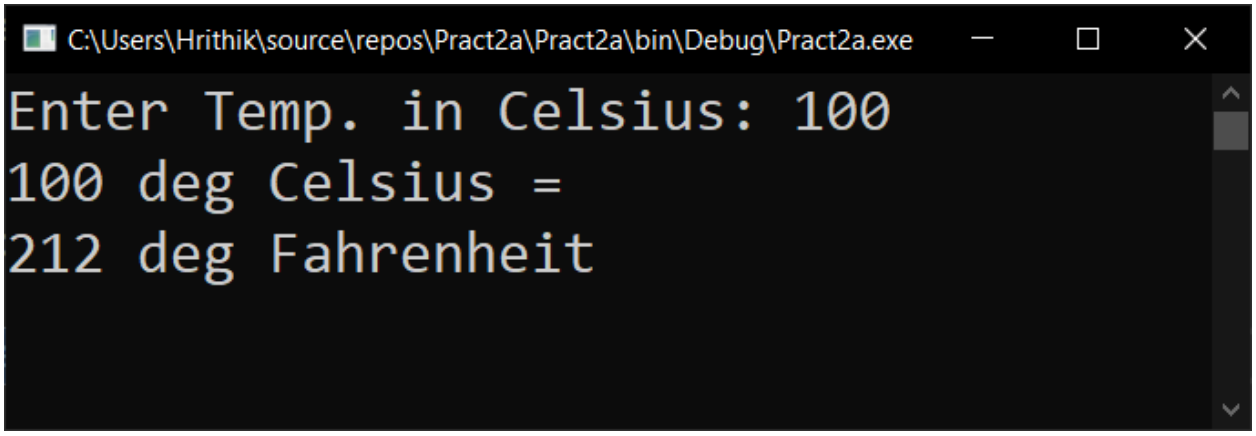
Output:



b) Source Code (Temperature Conversion):

```
using System;

namespace Pract2a {
    class Program {
        static void Main(string[] args) {
            Console.Write("Enter Temp. in Celsius: ");
            int cel = Int32.Parse(Console.ReadLine());
            Console.WriteLine(cel + " deg Celsius = \n" +
ctof(cel) + " deg Fahrenheit");
            Console.ReadKey();
        }
        static int ctof(int celsius) {
            return (celsius * 9) / 5 + 32;
        }
    }
}
```

Output:



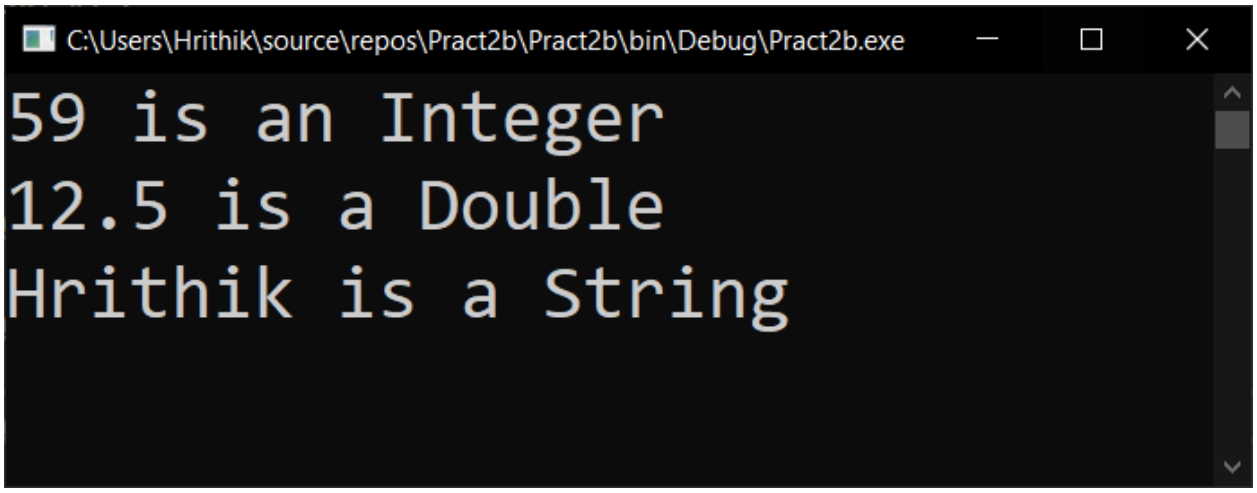
2b Aim: Create simple application to demonstrate use of following concepts.

a) Source Code (Function Overloading):

```
using System;

namespace Pract2b {
    class Program {
```

```
        static void Main(string[] args) {
            Console.WriteLine(type(59));
            Console.WriteLine(type(12.5));
            Console.WriteLine(type("Hrithik"));
            Console.ReadKey();
        }
        static String type(int a){return a + " is an Integer";}
        static String type(double a){return a + " is a Double";}
        static String type(String a){return a + " is a String";}
    }
}
```

Output:

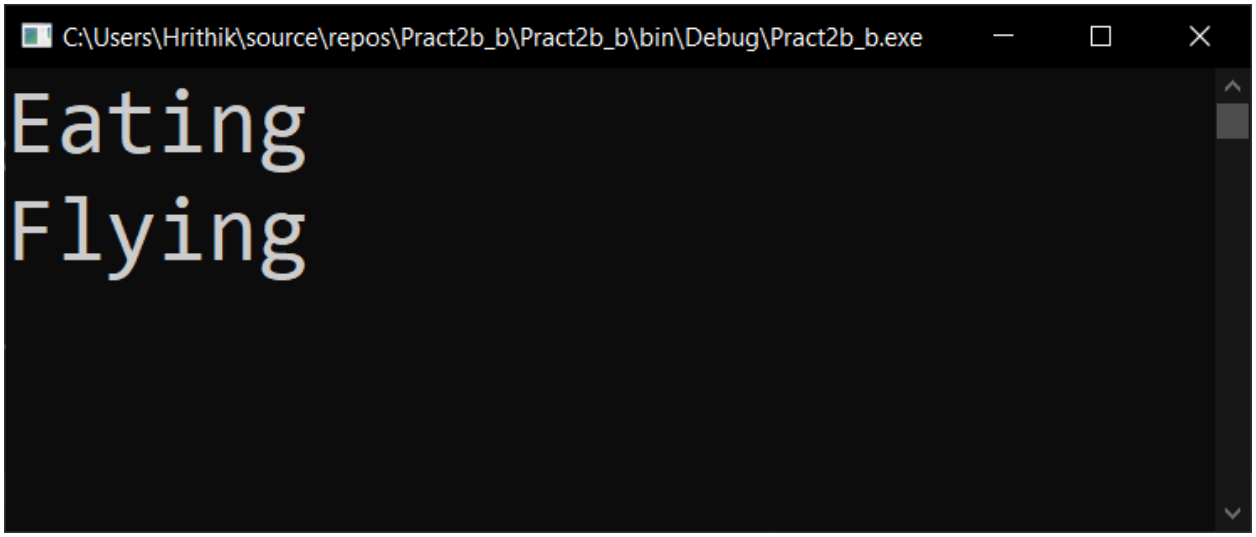


b1) Source Code (Single Inheritance):

```
using System;
namespace Pract2b_b {
    class Program {
        static void Main(string[] args) {
            Bird bird = new Bird();
            bird.eat();
            bird.fly();
            Console.ReadKey();
        }
    }
    class Creatures {
        public void eat() { Console.WriteLine("Eating"); }
    }
```

```
    class Bird : Creatures {
        public void fly() { Console.WriteLine("Flying"); }
    }
}
```

Output:

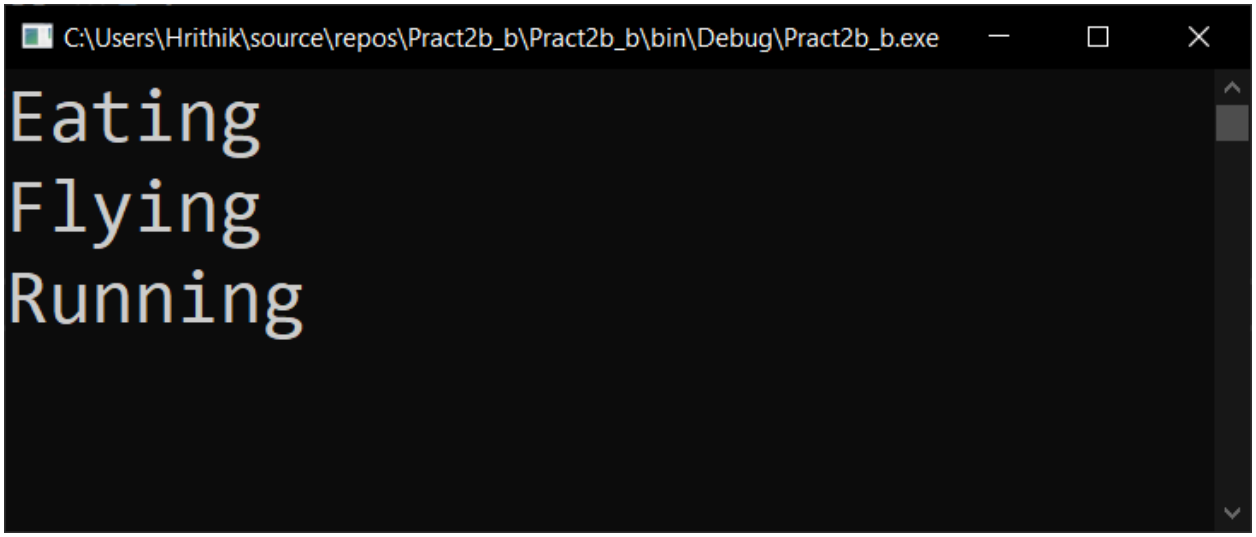


b2) Source Code (Multi Level Inheritance):

```
using System;
namespace Pract2b_b {
    class Program {
        static void Main(string[] args) {
            FlyingSquirrel fs = new FlyingSquirrel();
            fs.eat();
            fs.fly();
            fs.runOnLegs();
            Console.ReadKey();
        }
    }
    class Creatures {
        public void eat() { Console.WriteLine("Eating"); }
    }
    class Bird : Creatures {
        public void fly() { Console.WriteLine("Flying"); }
    }
    class FlyingSquirrel : Bird {
```

```
        public void runOnLegs(){ Console.WriteLine("Running"); }
    }
}
```

Output:

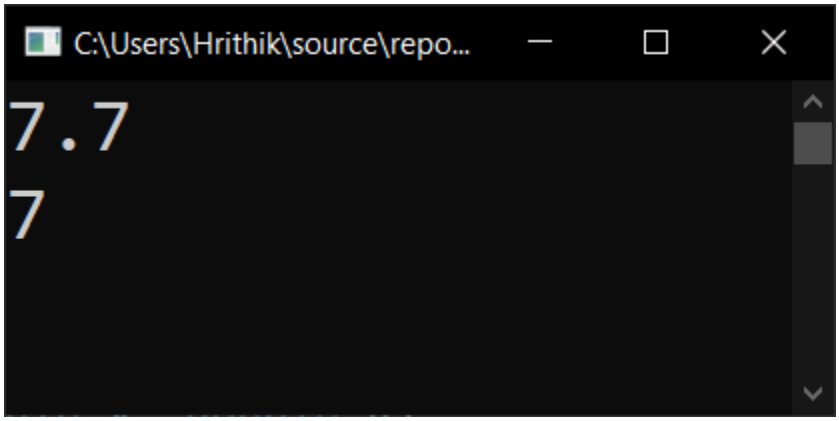


c) Source Code (Constructor Overloading):

```
using System;

namespace Pract2b_c {
    class Program {
        static void Main(string[] args) {
            Add add = new Add(2.1, 5.6);
            Add add2 = new Add(2, 5);
            Console.WriteLine(add.show());
            Console.WriteLine(add2.show());
            Console.ReadKey();
        }
    }
    class Add {
        double a, b;
        public Add(int x, int y) {
            this.a = x;
            this.b = y;
        }
        public Add(double x, double y) {
            this.a = x;
```

```
            this.b = y;
        }
        public double show() { return a + b; }
    }
}
```

Output:

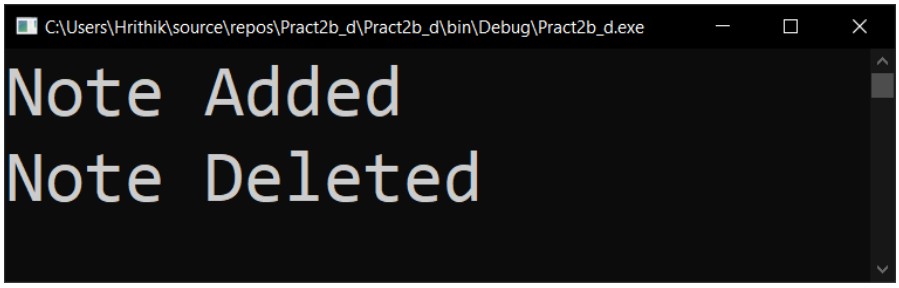


d) Source Code (Interfaces):

```
using System;

namespace Pract2b_d {
    class Program {
        static void Main(string[] args) {
            Note note = new Note();
            note.addNote();
            note.deleteNote();
            Console.ReadKey();
        }
    }
    interface INote {
        void addNote();
        void deleteNote();
    }
    class Note : INote {
        public void addNote() {
            Console.WriteLine("Note Added");
        }
        public void deleteNote() {
            Console.WriteLine("Note Deleted");
```

```
        }
    }
}
```

Output:

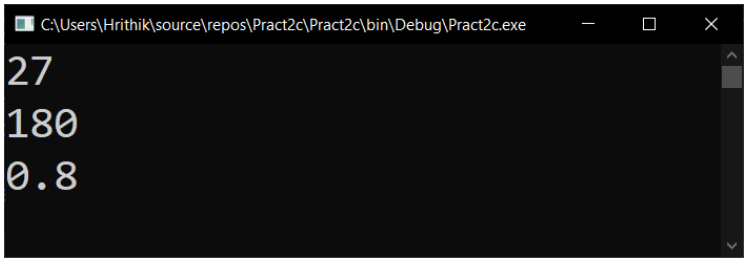


2c Aim: Create simple application to demonstrate use of following concepts.

a) Source Code (Using Delegates and events):

```
using System;

namespace Pract2c {
    class Program {
        delegate double Math(int a, int b);
        static void Main(string[] args) {
            Math math1 = new Math(add);
            Math math2 = new Math(mul);
            Math math3 = new Math(div);

            Console.WriteLine(math1(12, 15));
            Console.WriteLine(math2(12, 15));
            Console.WriteLine(math3(12, 15));

            Console.ReadKey();
        }
        static double add(int a, int b) { return a + b; }
        static double mul(int a, int b) { return a * b; }
        static double div(int a, int b) { return (double) a/b; }
    }
}
```

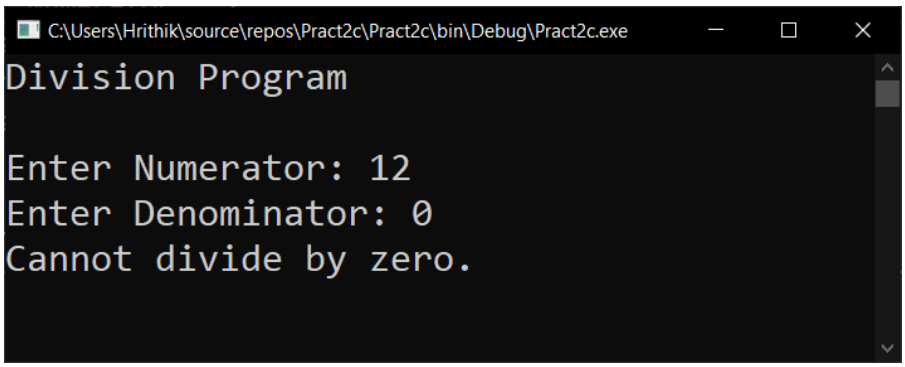
Output:



b) Source Code (Exception Handling):

```
using System;

namespace Pract2b_d {
    class Program {
        static void Main(string[] args) {
            Console.WriteLine("Division Program\n");
            Console.Write("Enter Numerator: ");
            int num = Int32.Parse(Console.ReadLine());
            Console.Write("Enter Denominator: ");
            int den = Int32.Parse(Console.ReadLine());
            try {
                Console.WriteLine(num+" / "+den+" = "+num/den);
            }
            catch (DivideByZeroException e) {
                Console.WriteLine("Cannot divide by zero.");
            }
            Console.ReadKey();
        }
    }
}
```

Output:



3. Working with Web Forms and Controls

3a Aim: Create a simple web page with various sever controls to demonstrate setting and use of their properties.

Source Code:

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Practical3a_AutoPostBack.WebForm1" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label1" runat="server" Text="Roll
No"></asp:Label>
    
            <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Label ID="Label2" runat="server"
Text="Name"></asp:Label>
       
            <asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Label ID="Label3" runat="server"
Text="Class"></asp:Label>
       
            <asp:RadioButton ID="RadioButton1" runat="server"
GroupName="class" Text="FY" />
 
```

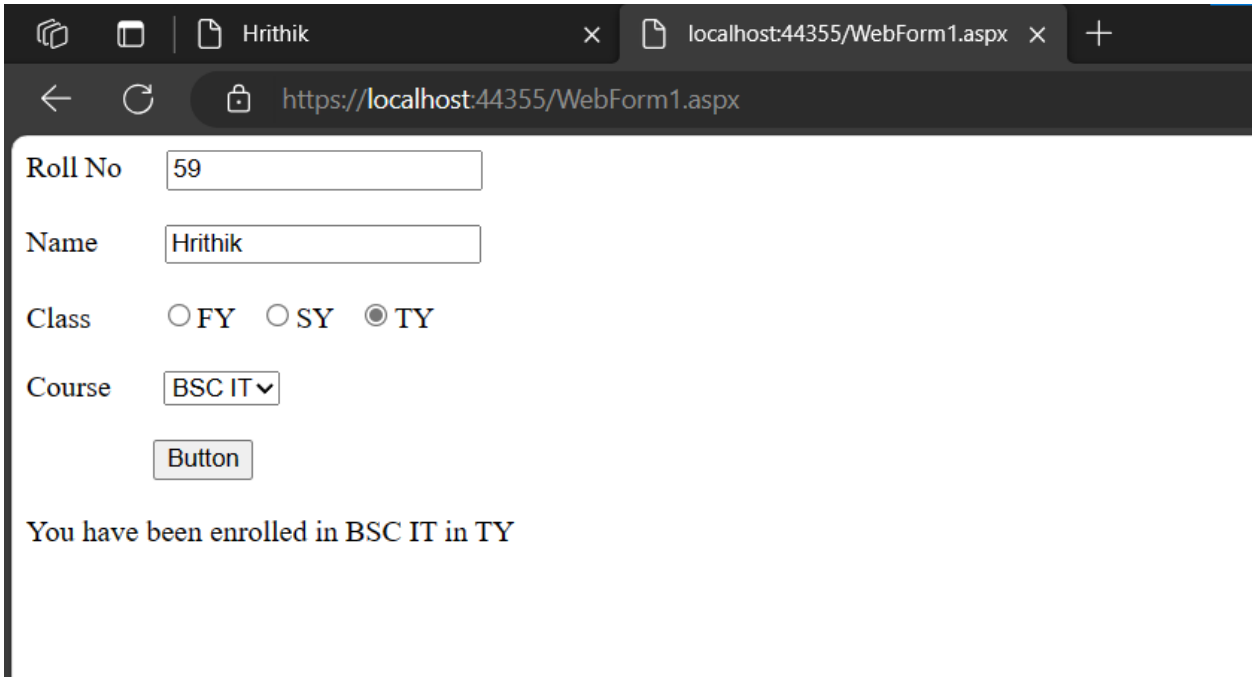
```
            <asp:RadioButton ID="RadioButton2" runat="server"
GroupName="class" Text="SY" />
 
            <asp:RadioButton ID="RadioButton3" runat="server"
GroupName="class" Text="TY" />
            <br />
            <br />
            <asp:Label ID="Label4" runat="server"
Text="Course"></asp:Label>
     
            <asp:DropDownList ID="DropDownList1" runat="server"
OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged">
                <asp:ListItem>BCOM</asp:ListItem>
                <asp:ListItem>BAF</asp:ListItem>
                <asp:ListItem>BSC IT</asp:ListItem>
                <asp:ListItem>BBI</asp:ListItem>
            </asp:DropDownList>
            <br /><br>
          &nbs
p;     
            <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Button" />
            <br />
            <br />
            <asp:Label ID="Label5" runat="server"
Text="Label"></asp:Label>
        </div>
    </form>
</body>

</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical3a_AutoPostBack{
```

```
    public partial class WebForm1 : System.Web.UI.Page {
       protected void Button1_Click(object sender, EventArgs e){
            string s;
            if (RadioButton1.Checked == true) {
                s = RadioButton1.Text;
            }
            if (RadioButton2.Checked == true) {
                s = RadioButton2.Text;
            }
            else {
                s = RadioButton3.Text;
            }
            Label5.Text += " in " + s;
        }

        protected void DropDownList1_SelectedIndexChanged(object
sender, EventArgs e) {
            Label5.Text = "You have been enrolled in " +
DropDownList1.SelectedItem;
        }
    }
}
```

Output:



3b Aim: Demonstrate the use of Calendar control to perform following operations.
a) Display messages in a calendar control
b) Display vacation in a calendar control
c) Selected day in a calendar control using style
d) Difference between two calendar dates.

Source Code:

a) WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Calendar.WebForm1" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Calendar ID="Calendar1" runat="server"
BackColor="#FFFF99" BorderColor="#003366" BorderStyle="Solid"
CellSpacing="2" ForeColor="#003399" NextPrevFormat="ShortMonth"
Width="10pt" OnDayRender="Calendar1_DayRender"
OnSelectionChanged="Calendar1_SelectionChanged"
DayNameFormat="Full" ShowGridLines="True">
                <DayHeaderStyle BackColor="Aqua" />
                <DayStyle BackColor="#FFFF66"
ForeColor="#003300" />
                <NextPrevStyle BackColor="#FF99FF" />
                <OtherMonthDayStyle BackColor="Aqua" />
                <SelectedDayStyle BackColor="#FFCCCC" />
                <TitleStyle BorderStyle="Dotted" />
            </asp:Calendar>
            <br />
        </div>
    </form>
</body>

</html>
```

a) WebForm1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Calendar
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Calendar1_DayRender(object sender,
DayRenderEventArgs e) {
            if (e.Day.Date.Day == 9) {
                e.Cell.Controls.Add(new
LiteralControl("<br>Holiday"));
            }
            if (e.Day.Date >= new DateTime(2023, 09, 20) &&
(e.Day.Date <= new
           DateTime(2023, 09, 25))) {
                e.Cell.BackColor = System.Drawing.Color.Red;
                e.Cell.BorderColor =
System.Drawing.Color.NavajoWhite;
                e.Cell.BorderWidth = new Unit(3);
                if (e.Day.IsOtherMonth) {
                    e.Cell.Controls.Clear();
                }
            }
        }
    }
}
```

b) WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Calendar.WebForm1" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
```

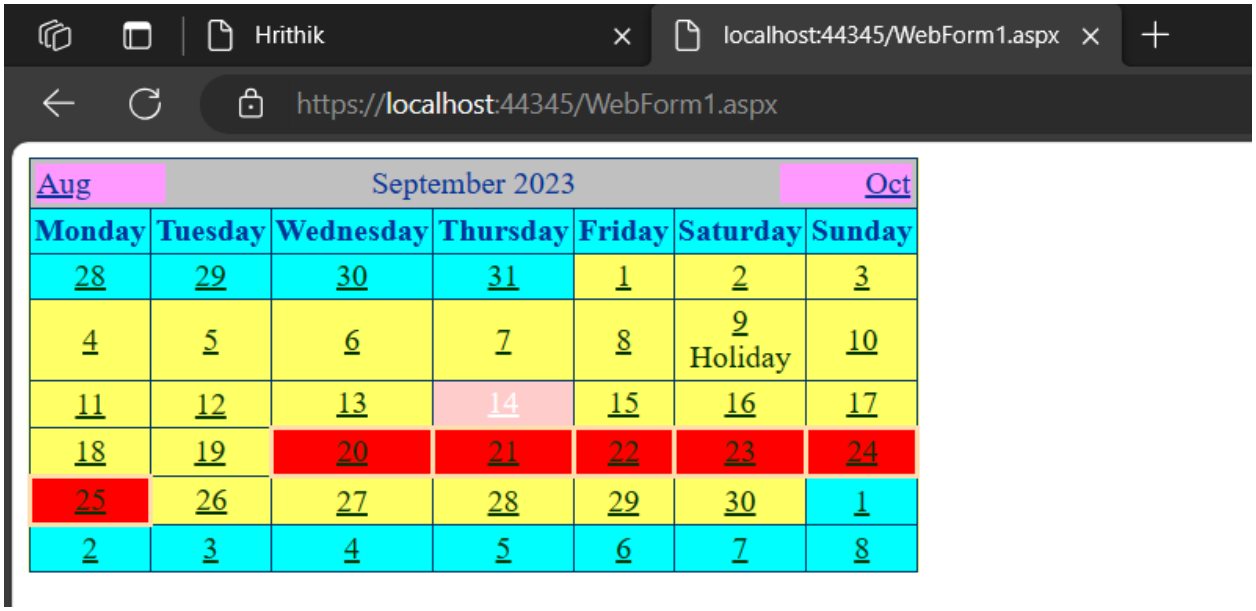
```
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Calendar ID="Calendar2"
runat="server"></asp:Calendar>
            <br />
            <asp:Calendar ID="Calendar3"
runat="server"></asp:Calendar>
            <br />
            <asp:Label ID="Label1" runat="server"
Text=""></asp:Label>
 <br />
            <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Calculate Number of Days" />
            <br />

        </div>
    </form>
</body>
</html>
```

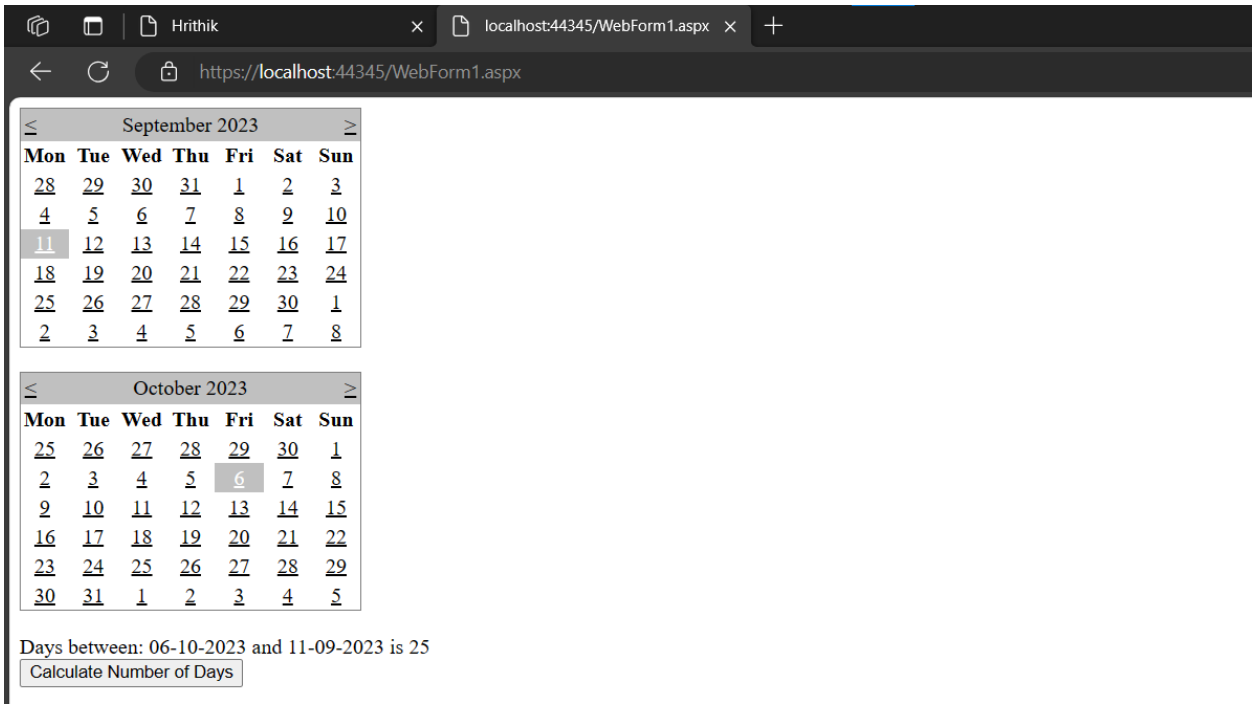
b) WebForm1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Calendar {
    public partial class WebForm1 : System.Web.UI.Page{

       protected void Button1_Click(object sender, EventArgs e){
            TimeSpan t = Calendar3.SelectedDate -
Calendar2.SelectedDate;
            Label1.Text = "Days between: " +
Calendar3.SelectedDate.ToShortDateString() + " and "+
Calendar2.SelectedDate.ToShortDateString() + " is " +
t.Days.ToString();
```

```
            }
        }

}
```

Output:


Hrithik
localhost:44345/WebForm1.aspx
https://localhost:44345/WebForm1.aspx
Aug
September 2023
Oct
Monday Tuesday Wednesday Thursday Friday Saturday Sunday
28 29 30 31 1 2 3
4 5 6 7 8 9 Holiday 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 1
2 3 4 5 6 7 8


Hrithik
localhost:44345/WebForm1.aspx
https://localhost:44345/WebForm1.aspx
≤ September 2023 ≥
Mon Tue Wed Thu Fri Sat Sun
28 29 30 31 1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 1
2 3 4 5 6 7 8
≤ October 2023 ≥
Mon Tue Wed Thu Fri Sat Sun
25 26 27 28 29 30 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31 1 2 3 4 5
Days between: 06-10-2023 and 11-09-2023 is 25
Calculate Number of Days

3c Aim: Demonstrate the use of Treeview control perform following operations. a) Treeview control and datalist b) Treeview operations.

Source Code:

Webform1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Practical3b_TreeView.WebForm1" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TreeView ID="TreeView1" runat="server"
OnSelectedNodeChanged="TreeView1_SelectedNodeChanged"
OnTreeNodeCollapsed="TreeView1_TreeNodeCollapsed">
                <Nodes>
                    <asp:TreeNode Text="BCom" Value="BCom"
ShowCheckBox="True">
                        <asp:TreeNode Text="FYBCom"
Value="FYBCom" ShowCheckBox="True"></asp:TreeNode>
                        <asp:TreeNode Text="SYBCom"
Value="SYBCom" ShowCheckBox="True"></asp:TreeNode>
                        <asp:TreeNode Text="TYBCom"
Value="TYBCom" ShowCheckBox="True"></asp:TreeNode>
                    </asp:TreeNode>
                    <asp:TreeNode Text="BScIT" Value="BScIT"
ShowCheckBox="True">
                        <asp:TreeNode Text="FYIT" Value="FYIT"
ShowCheckBox="True"></asp:TreeNode>
                        <asp:TreeNode Text="SYIT" Value="SYIT"
ShowCheckBox="True"></asp:TreeNode>
                        <asp:TreeNode Text="TYIT" Value="TYIT"
ShowCheckBox="True"></asp:TreeNode>
                    </asp:TreeNode>
                </Nodes>
```

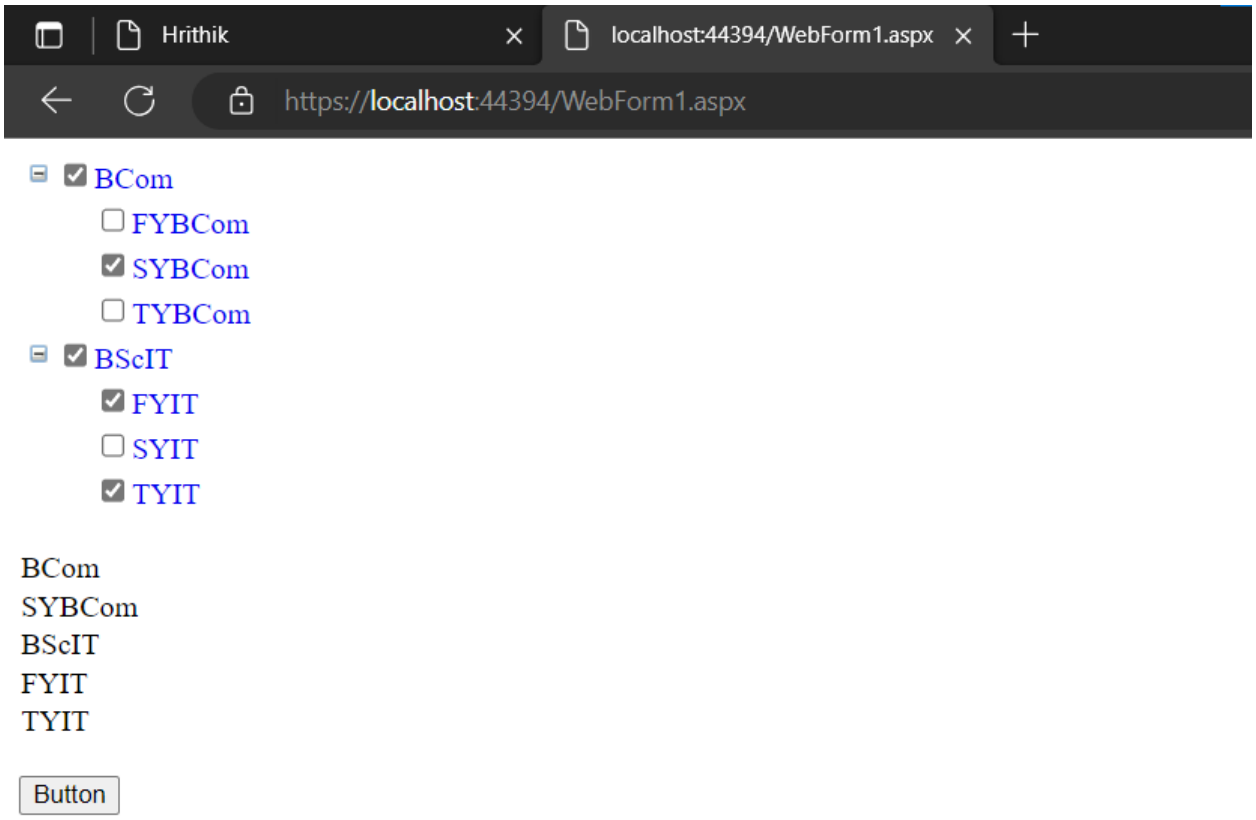
```
            </asp:TreeView>
            <br />
            <asp:DataList ID="DataList1" runat="server">
                <ItemTemplate>
                    <%# Eval("text") %>
                </ItemTemplate>
            </asp:DataList>
            <br />
            <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Button" />
            <br />
        </div>
    </form>
</body>
</html>
```

Webform1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical3b_TreeView {
    public partial class WebForm1 : System.Web.UI.Page {
        protected void Button1_Click(object sender, EventArgs e)
{
            TreeNodeCollection T;
            T = TreeView1.CheckedNodes;

            DataList1.DataSource = T;
            DataList1.DataBind();
            DataList1.Visible = true;
        }
        protected void TreeView1_SelectedNodeChanged(object
sender, EventArgs e) {
            Response.Write("Selection Option: " +
TreeView1.SelectedValue);
        }
```

```
        protected void TreeView1_TreeNodeCollapsed(object
sender, TreeNodeEventArgs e) {
            Response.Write("Value Collapsed: " + e.Node.Value);
        }
    }
}
```

Output:



4. Working with Form Controls

4a Aim: Create a Registration form to demonstrate use of various Validation controls.

Source Code:

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Practcal4a_Validation.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Enter Name:
            <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="*Name Required"
ForeColor="Red"></asp:RequiredFieldValidator>
            <br /><br />
            Enter Password:
            <asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator2" runat="server"
ControlToValidate="TextBox2" ErrorMessage="*Password Required"
ForeColor="Red"></asp:RequiredFieldValidator>
            <br /><br />
            Confirm Password: <asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
```

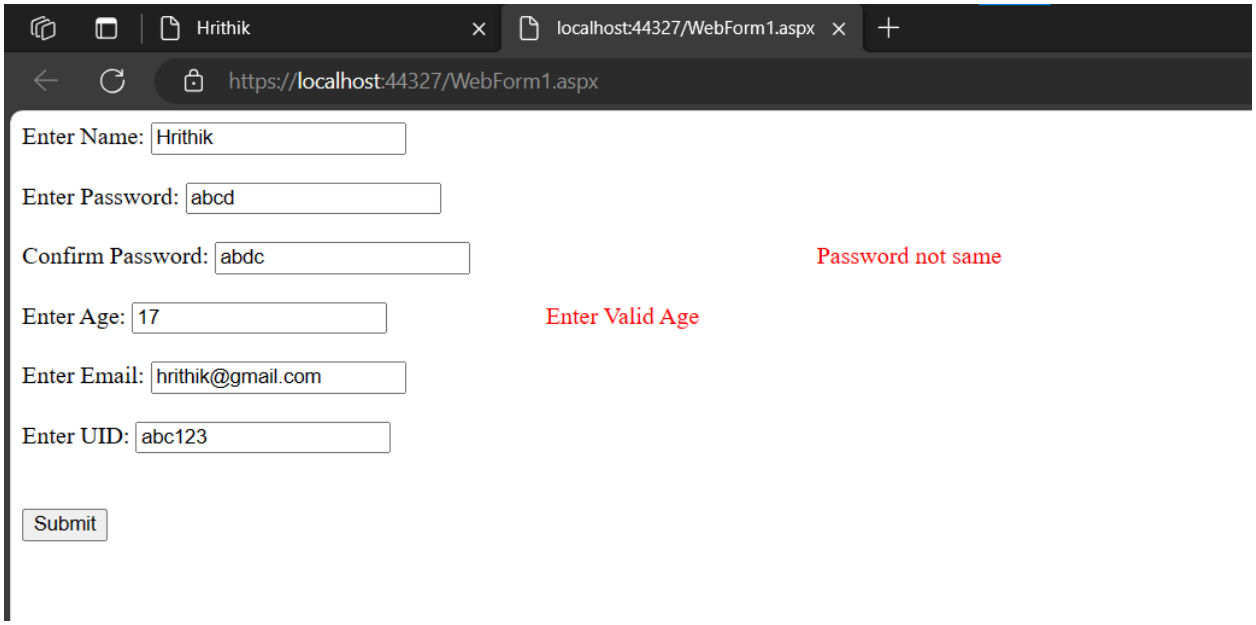
```
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator3" runat="server"
ControlToValidate="TextBox3" ErrorMessage="*Password
Confirmation Required"
ForeColor="Red"></asp:RequiredFieldValidator>
            <asp:CompareValidator ID="CompareValidator1"
runat="server" ErrorMessage="Password not same" ForeColor="Red"
ControlToCompare="TextBox2"
ControlToValidate="TextBox3"></asp:CompareValidator>
            <br /><br />
            Enter Age: <asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator4" runat="server"
ControlToValidate="TextBox4" ErrorMessage="*Age Required"
ForeColor="Red"></asp:RequiredFieldValidator>
            <asp:RangeValidator ID="RangeValidator1"
runat="server" ErrorMessage="Enter Valid Age" ForeColor="Red"
MaximumValue="25" MinimumValue="19"
ControlToValidate="TextBox4"></asp:RangeValidator>
            <br /><br />
            Enter Email: <asp:TextBox ID="TextBox5"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator5" runat="server"
ControlToValidate="TextBox5" ErrorMessage="*Email Required"
ForeColor="Red"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator
ID="RegularExpressionValidator1" runat="server"
ErrorMessage="Enter Email in correct format" ForeColor="Red"
ControlToValidate="TextBox5"
ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+
)*"></asp:RegularExpressionValidator>
            <br /><br />
            Enter UID: <asp:TextBox ID="TextBox6"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator6" runat="server"
ControlToValidate="TextBox6" ErrorMessage="*UID Required"
ForeColor="Red"></asp:RequiredFieldValidator>
            <br /><br /><br />
```

```
            <asp:Button ID="Button1" runat="server"
Text="Submit" OnClick="Button1_Click" />
        </div>
    </form>
</body>
</html>
```

Note:
Write this in Web.config file under configuration tag if error in output:

```
<appSettings>
     <add key="ValidationSettings:UnobtrusiveValidationMode"
value="None" />
</appSettings>
```

Output:



4b Aim: Create Web Form to demonstrate use of Adrotator Control.

Source Code:

WebForm1.aspx

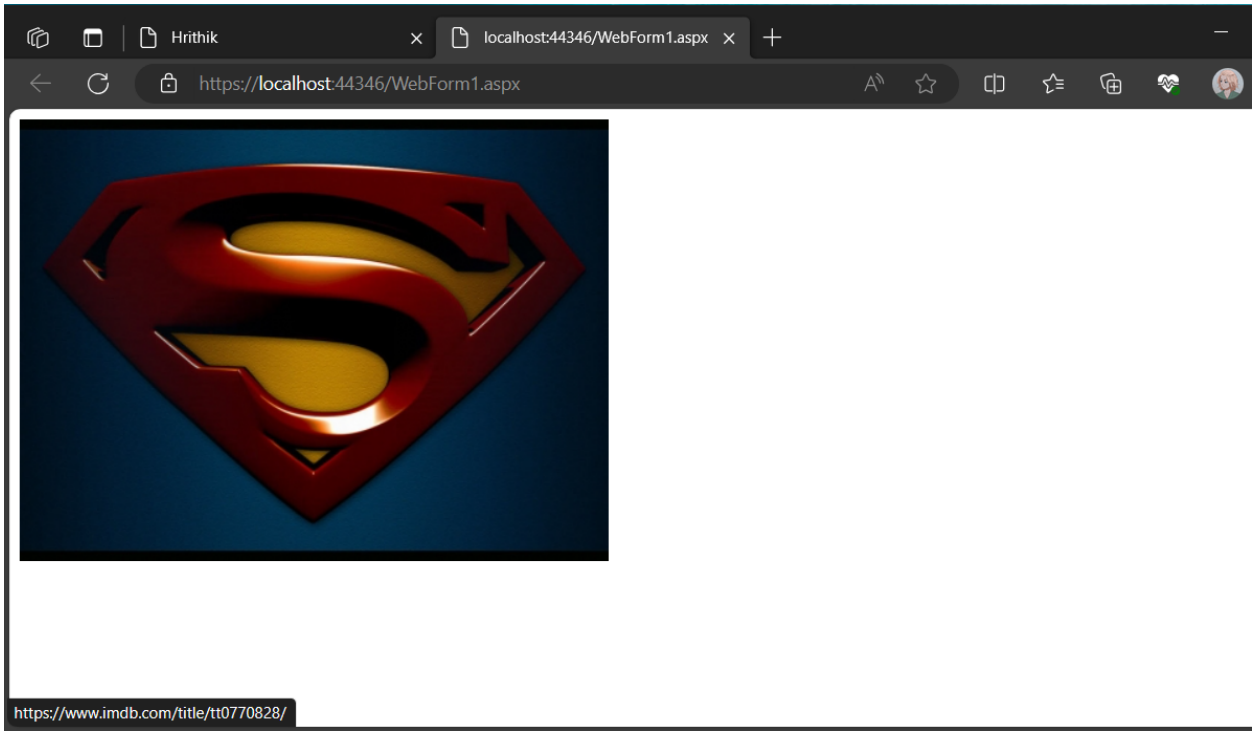
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Practical4b_Adrotator.WebForm1" %>
```

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:AdRotator ID="AdRotator1" runat="server"
DataSourceID="XmlDataSource1" />
            <asp:XmlDataSource ID="XmlDataSource1"
runat="server" DataFile="~/XMLFile1.xml"></asp:XmlDataSource>
        </div>
    </form>
</body>
</html>
```

XML File:

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
     <Ad>
          <ImageUrl>images/batman.jpg</ImageUrl>

<NavigateUrl>https://www.imdb.com/title/tt1877830/</NavigateUrl>
          <AlternateText>Batman Logo</AlternateText>
          <Impressions>10</Impressions>
          <Keywords>Batman Dc Dceu imbatman</Keywords>
     </Ad>
     <Ad>
          <ImageUrl>images/superman.jpg</ImageUrl>

<NavigateUrl>https://www.imdb.com/title/tt0770828/</NavigateUrl>
          <AlternateText>Superman Logo</AlternateText>
          <Impressions>10</Impressions>
          <Keywords>super man superman dc dceu</Keywords>
     </Ad>

</Advertisements>
```

Output:



4c Aim: Create Web Form to demonstrate use of User Controls.

Source Code:

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Practical4c_UserControls.WebForm1" %>
<%@ Register TagPrefix="uc1" TagName="WebUserControl"
Src="WebUserControl1.ascx" %>
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
```

```
            <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
            <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Button" />
            <asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
            <br>
            <uc1:WebUserControl runat="server"
ID="WebUserControl1" />
        </div>
    </form>
</body>
</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical4c_UserControls {
    public partial class WebForm1 : System.Web.UI.Page {
       protected void Button1_Click(object sender, EventArgs e){
            TextBox2.Text = "Happy FS Day " + TextBox1.Text;
        }
    }
}
```

WebUserControl1.ascx

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeBehind="WebUserControl1.ascx.cs"
Inherits="Practical4c_UserControls.WebUserControl1" %>

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

<asp:Button ID="Button1" runat="server" Text="Button"
OnClick="Button1_Click" />

<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
```

WebUserControl1.ascx.cs

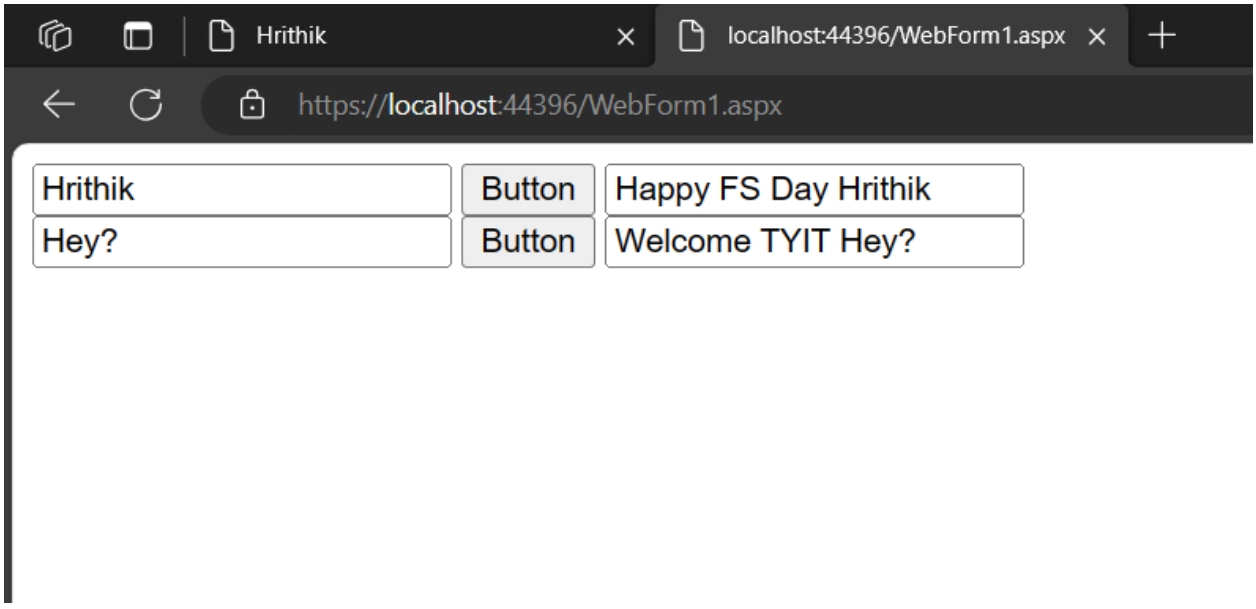
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical4c_UserControls {
    public partial class WebUserControl1 :
System.Web.UI.UserControl {
        protected void Button1_Click(object sender, EventArgs e){
             TextBox2.Text = "Welcome TYIT " + TextBox1.Text;
         }
    }
}
```

Note:

Don't forget to add this at top above doctype tag in WebForm1.aspx:

<%@ Register TagPrefix="uc1" TagName="WebUserControl" Src="WebUserControl1.ascx" %>

Output:



5. Working with Navigation, Beautification and Master page.

5a Aim: Create Web Form to demonstrate use of Website Navigation controls and Site Map.

Source Code:

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Practical5a_SiteMap.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:SiteMapPath ID="SiteMapPath1" runat="server">
            </asp:SiteMapPath>
            <br />
            <br />
            <asp:Menu ID="Menu1" runat="server"
DataSourceID="SiteMapDataSource1">
            </asp:Menu>
            <br />
            <asp:SiteMapDataSource ID="SiteMapDataSource1"
runat="server" />
            <br />
        </div>
    </form>
</body>

</html>
```

WebForm2.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs"
Inherits="Practical5a_SiteMap.WebForm2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:SiteMapPath ID="SiteMapPath1" runat="server">
            </asp:SiteMapPath>
            <br /> Welcome to Desired Movies!
            <br />
        </div>
    </form>
</body>

</html>
```

WebForm3.aspx

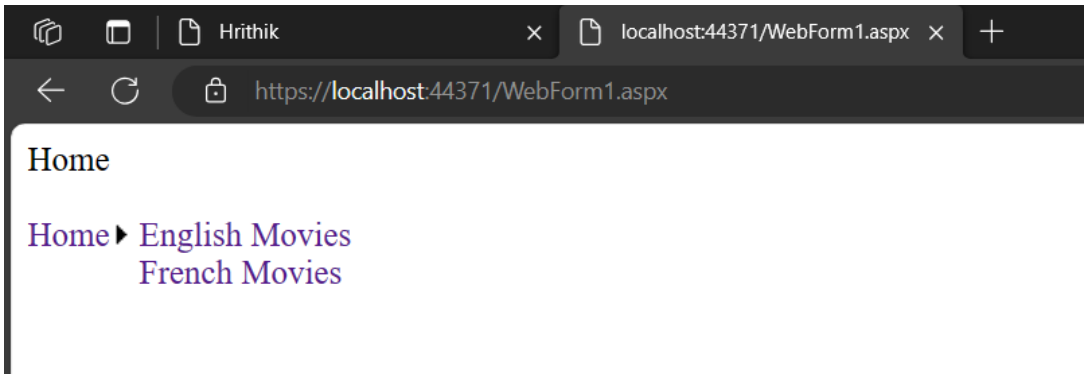
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm3.aspx.cs"
Inherits="Practical5a_SiteMap.WebForm3" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:SiteMapPath ID="SiteMapPath1" runat="server">
            </asp:SiteMapPath>
            <br /> Welcome to Desired Movies: French
```

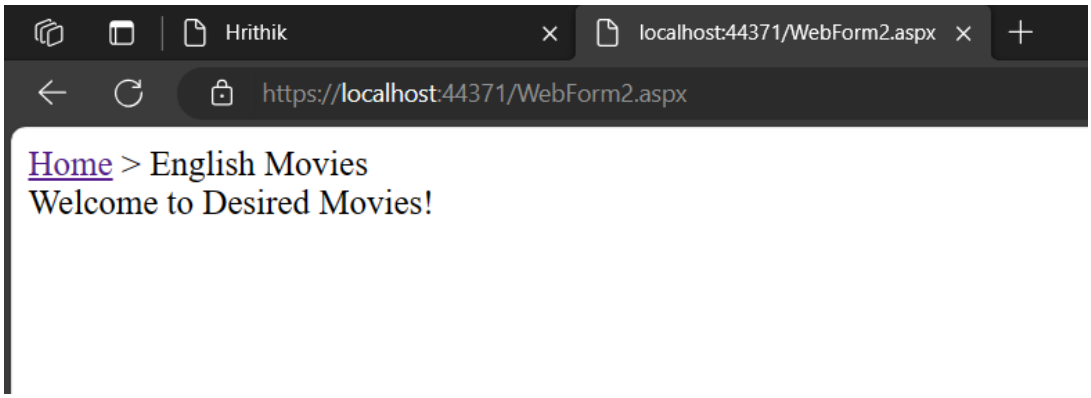
```
            <br />
        </div>
    </form>
</body>

</html>
```

Web.sitemap

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap
xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
     <siteMapNode url="WebForm1.aspx" title="Home"
description="Desired Movies Home">
          <siteMapNode url="WebForm2.aspx" title="English
Movies"  description="Second Page" />
          <siteMapNode url="WebForm3.aspx" title="French Movies"
description="Third Page" />
     </siteMapNode>
</siteMap>
```

Output:





5b Aim: Create a web application to demonstrate use of Master Page with applying Styles and Themes for page beautification.

Source Code:

Site1.Master

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeBehind="Site1.master.cs" Inherits="pract5b.Site1" %>

<!DOCTYPE html>
<html>
<head runat="server">
     <title></title>
     <asp:ContentPlaceHolder ID="head" runat="server">
     </asp:ContentPlaceHolder>
</head>
<body>
     <link href="StyleSheet1.css" rel="stylesheet"
type="text/css" />
     <form id="form1" runat="server">
        <div>
            <asp:ContentPlaceHolder ID="ContentPlaceHolder1"
runat="server">
            </asp:ContentPlaceHolder>
        </div>
     </form>
</body>
</html>
```

StyleSheet1.css

```
body {
     background-color:aqua;
     font:italic;
}
```

Skin1.skin

```
<asp:Label runat="server" SkinId="lb1" backcolor=yellow/>
```

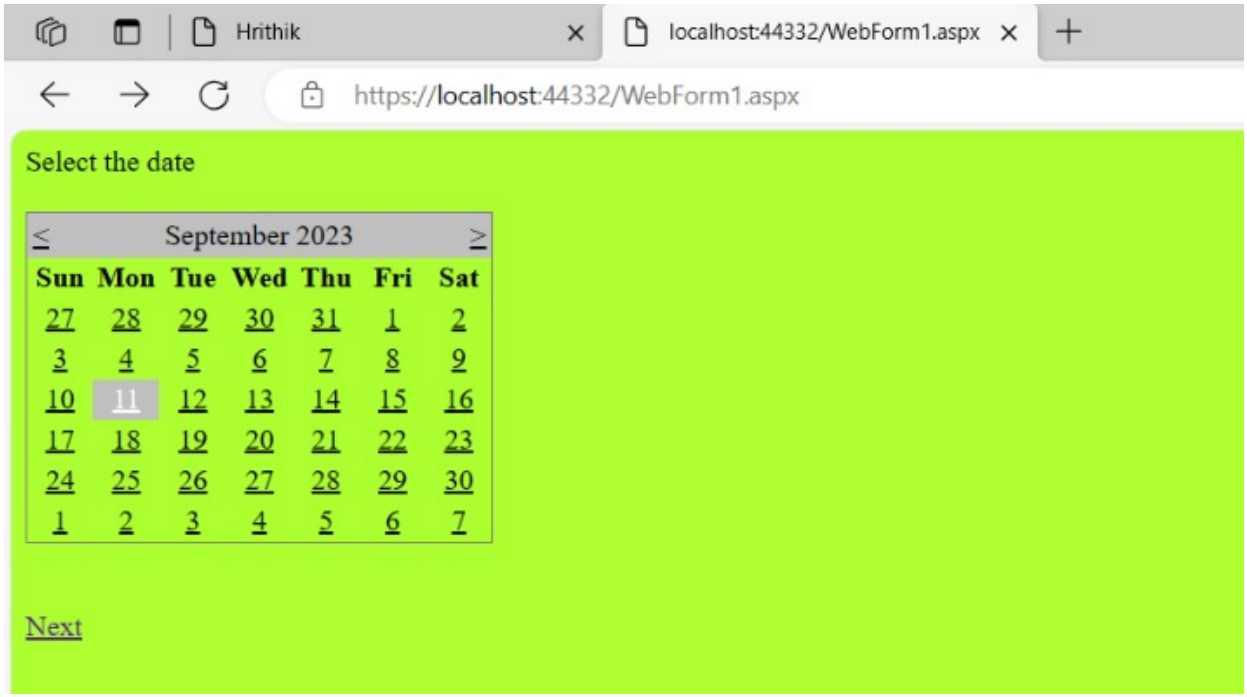
WebForm1.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master"
AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="pract5b.WebForm1" Theme="Skin1"%>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
     <asp:Label ID="Label1" runat="server" SkinId="lb1"
Text="Select the date"></asp:Label>
     <br />
     <br />
     <asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>
     <br />
     <br />
     <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/WebForm2.aspx">Next</asp:HyperLink>
     <br />
</asp:Content>
```

WebForm2.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master"
AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="pract5b.WebForm2" Theme="Skin1"%>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
     <asp:Label ID="Label1" runat="server" Text="Label"
SkinId="lb1"></asp:Label>
 
     <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
</asp:Content>
```

Output:



5c Aim: Create a web application to demonstrate various states of ASP.NET Pages.

Source Code:

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Practical4c_States.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:HiddenField ID="HiddenField1" runat="server"
Value="2" />
            <br />
```

```
            <asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
            <br />
            <br />
            <asp:Label ID="Label2" runat="server"
Text="Label"></asp:Label>
            <br />
            <br />
            <asp:Button ID="Button1" runat="server" Text="View
State" OnClick="Button1_Click" />
            <br />
            <br />
            <asp:Button ID="Button2" runat="server" Text="Hidden
Field" OnClick="Button2_Click" />
            <br />
            <br />
            <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Button ID="Button3" runat="server"
Text="Cookies" OnClick="Button3_Click" style="height: 26px" />
            <br />
        </div>
    </form>
</body>

</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical4c_States{
    public partial class WebForm1 : System.Web.UI.Page {
        protected void Page_Load(object sender, EventArgs e){
            if (IsPostBack) {
```

```
                if (ViewState["count"] != null) {
                    int Viewstateval =
Convert.ToInt32(ViewState["count"]) + 1;
                    Label2.Text = "view state: " +
Viewstateval.ToString();
                    ViewState["count"] =
Viewstateval.ToString();
                }
                else {
                    ViewState["count"] = "1";
                }
            }
        }

        protected void Button1_Click(object sender, EventArgs e){
            Label1.Text = ViewState["count"].ToString();
        }

        protected void Button2_Click(object sender, EventArgs e){
            int val = Convert.ToInt32(HiddenField1.Value) + 1;
            HiddenField1.Value = val.ToString();
        }

        protected void Button3_Click(object sender, EventArgs e){
            HttpCookie cookie = new HttpCookie("name");
            cookie.Value = TextBox1.Text;
            Response.Cookies.Add(cookie);
            Response.Redirect("WebForm2.aspx");
        }

    }
}
```

WebForm2.aspx

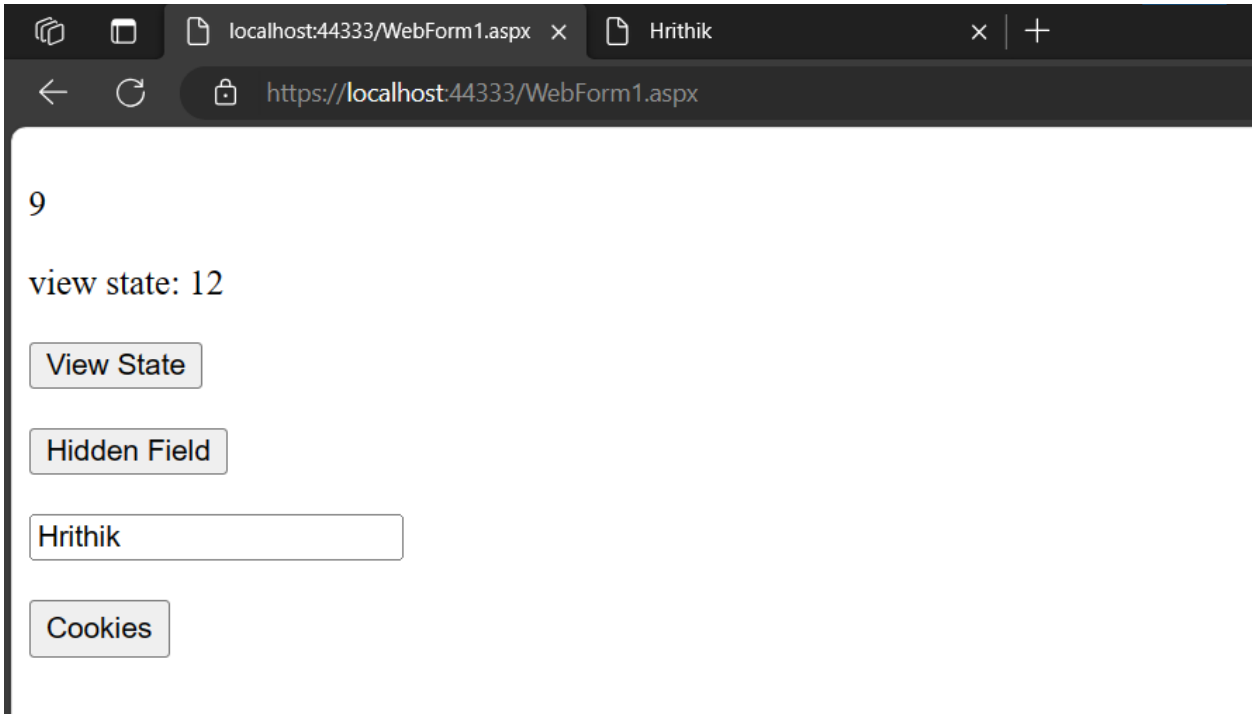
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs"
Inherits="Practical4c_States.WebForm2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label1" runat="server" Text="Label
"></asp:Label>
            <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
            <br />
        </div>
    </form>
</body>
</html>
```

WebForm2.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical4c_States{
    public partial class WebForm2 : System.Web.UI.Page {
        protected void Page_Load(object sender, EventArgs e) {
            if (Request.Cookies["name"] != null) {
                Response.Write("Welcome " +
Request.Cookies["name"].Value);
            }
        }
    }
}
```

Output:


localhost:44333/WebForm1.aspx
Hrithik
https://localhost:44333/WebForm1.aspx
9
view state: 12
View State
Hidden Field
Hrithik
Cookies

6. Working with Database.

6a Aim: Create a web application bind data in a multiline textbox by querying in another textbox.

Source Code:

WebForm1.aspx:

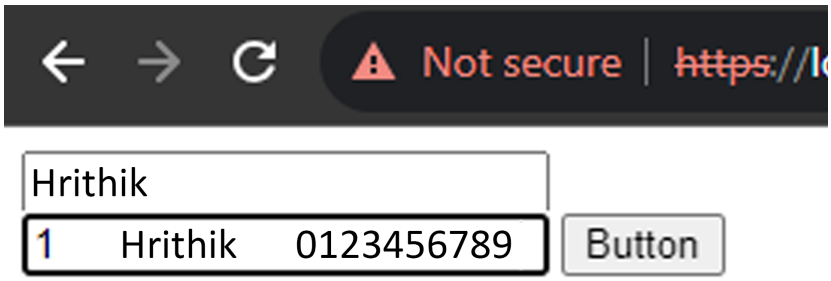
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Practical6_A_.WebForm1"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head runat="server">
        <title></title>
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
                <br />
                <asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
                <asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:dboConnectionString %>" SelectCommand="SELECT
* FROM [emp]">
                </asp:SqlDataSource>
                <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click1" Text="Button" />
            </div>
        </form>
    </body>
</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
namespace Practical6_A_ {
    public partial class WebForm1: System.Web.UI.Page {
        SqlConnection cn = new SqlConnection("Data
Source=HP;Initial Catalog=dbo;Integrated Security=True");
        SqlCommand co = new SqlCommand();
        SqlDataReader ds;
        protected void Page_Load(object sender, EventArgs e) {
            cn.Open();
            co.Connection = cn;
        }
        protected void TextBox1_TextChanged(object sender,
EventArgs e) {}
        protected void Button1_Click1(object sender, EventArgs
e) {
            co.CommandText = "select * from emp where name= '" +
TextBox1.Text + "';";
            ds = co.ExecuteReader();
            while (ds.Read()) {
                TextBox2.Text += ds[0].ToString() + "\t" +
ds[1].ToString() + "\t" + ds[2].ToString() + "\n";
            }
        }
    }
}
```

Output:



6b Aim: Create a web application to display records by using database.

Source Code:

WebForm1.aspx

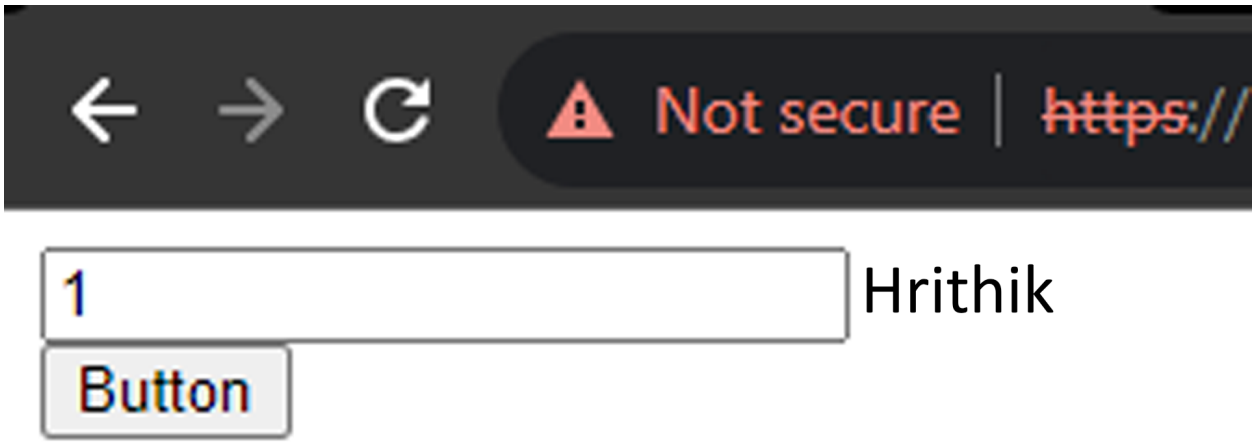
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Practical6_B_.WebForm1"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head runat="server">
        <title></title>
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
                <asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
                <br />
                <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click1" Text="Button" />
                <asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:db1ConnectionString %>" SelectCommand="SELECT
* FROM [student]">
                </asp:SqlDataSource>
            </div>
        </form>
    </body>
</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
```

```
namespace Practical6_B_ {
    public partial class WebForm1: System.Web.UI.Page {
        SqlConnection cn = new SqlConnection("Data
Source=HP;Initial Catalog=db1;Integrated Security=True");
        SqlCommand co = new SqlCommand();
        protected void Page_Load(object sender, EventArgs e) {
            cn.Open();
            co.Connection = cn;
        }
        protected void Button1_Click1(object sender, EventArgs
e) {
            co.CommandText = "select name from student where
sno= '" + TextBox1.Text + "';";
            Label1.Text = co.ExecuteScalar().ToString();
        }
    }
}
```

Output:



6c Aim: Demonstrate the use of Datalist link control.

Source Code:

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="practical6c.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
    <head runat="server">
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                <asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:db1ConnectionString %>" SelectCommand="SELECT
* FROM [Stud]">
                </asp:SqlDataSource> <br /> <br />
                <asp:DataList ID="DataList1" runat="server"
DataSourceID="SqlDataSource1">
                    <ItemTemplate> rollno:
                        <asp:Label ID="rollnoLabel"
runat="server" Text='
                                    <%# Eval("rollno") %>' />
                        <br /> name:
                        <asp:Label ID="nameLabel" runat="server"
Text='
                                        <%# Eval("name") %>' />
                        <br /> <br />
                    </ItemTemplate> </asp:DataList> </div>
        </form> </body> </html>
```

Output:

rollno: 1
name: Hrithik

rollno: 2
name: Sachin

rollno: 3
name: Faaiz

7. Working with Database.

7a Aim: Create a web application to display Databinding using dropdownlist control.

Source Code:

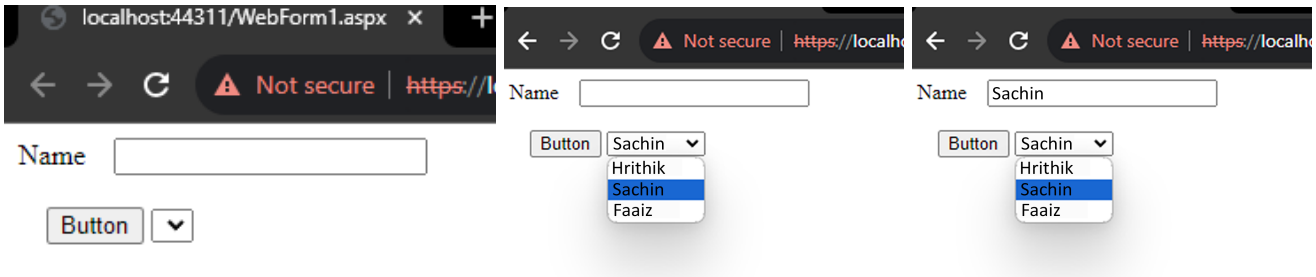
WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Practical7_a.WebForm1"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head runat="server">
        <title></title>
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                <asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>    <asp:TextBox
ID="TextBox1" runat="server"></asp:TextBox>
                <asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:db1ConnectionString %>" SelectCommand="SELECT
* FROM [Stud]">
                </asp:SqlDataSource>
                <br />
                <br />    
                <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Button" />
                <asp:DropDownList ID="DropDownList1"
runat="server"
OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged"></as
p:DropDownList>
            </div>
        </form>
    </body>
</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
namespace Practical7_a {
    public partial class WebForm1: System.Web.UI.Page {
        SqlConnection cn = new SqlConnection("Data
Source=HP;Initial Catalog=db1;Integrated Security=True");
        SqlCommand co = new SqlCommand();
        SqlDataReader ds;
        protected void Page_Load(object sender, EventArgs e) {
            cn.Open();
            co.Connection = cn;
        }
        protected void Button1_Click(object sender, EventArgs e)
{
            co.CommandText = "select * from stud";
            ds = co.ExecuteReader();
            DropDownList1.DataSource = ds;
            DropDownList1.DataTextField = "name";
            DropDownList1.DataBind();
        }
        protected void DropDownList1_SelectedIndexChanged(object
sender, EventArgs e) {
            TextBox1.Text = DropDownList1.Text;
        }
    }
}
```

Output:



7b Aim: Create a web application to display the phone no of an author using database.

Source Code:

WebForm1.aspx:

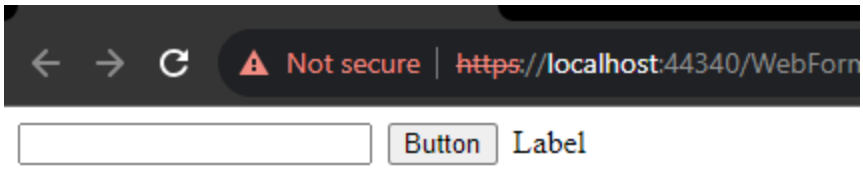
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Practical7_B.WebForm1"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head runat="server">
        <title></title>
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>  
                <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Button" />   <asp:Label
ID="Label1" runat="server" Text="Label"></asp:Label>
            </div>
        </form>
    </body>
</html>
```

WebForm1.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
namespace Practical7_B {
    public partial class WebForm1: System.Web.UI.Page {
        SqlConnection con = new SqlConnection("Data Source = HP;
Initial Catalog = db1; Integrated Security = True");
        SqlCommand co = new SqlCommand();
        SqlDataReader ds;
        protected void Page_Load(object sender, EventArgs e) {
```

```
            con.Open();
            co.Connection = con;
        }
        protected void Button1_Click(object sender, EventArgs e)
{
            co.CommandText = "select phone from emp where name =
'" + TextBox1.Text + "' ;";
            Label1.Text = co.ExecuteScalar().ToString();
        }
    }
}
```

Output:



7c Aim: Create a web application for inserting and deleting record from a database. (Using Execute-Non Query).

Source Code:

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Practical7_c.WebForm1"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head runat="server">
        <title></title>
    </head>
    <body>
        <form id="form1" runat="server">
```

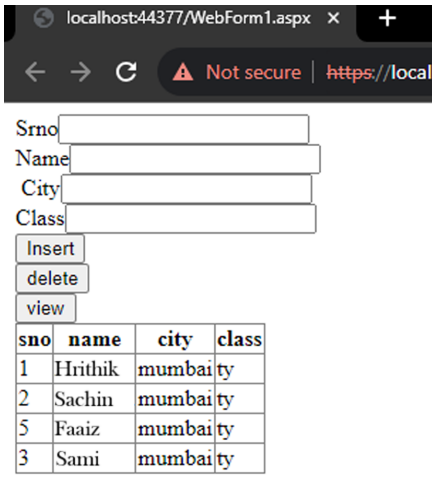
```
            <div> Srno <asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox>
                <br /> Name <asp:TextBox ID="TextBox2"
runat="server"></asp:TextBox>
                <br />  City <asp:TextBox ID="TextBox3"
runat="server"></asp:TextBox>
                <br /> Class <asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>
                <br />
                <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Insert" />
                <br />
                <asp:Button ID="Button2" runat="server"
OnClick="Button2_Click" Text="delete" />
                <br />
                <asp:Button ID="Button3" runat="server"
OnClick="Button3_Click" Text="view" />
                <asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:db1ConnectionString %>" SelectCommand="SELECT
* FROM [student]">
                </asp:SqlDataSource>
                <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False" DataSourceID="SqlDataSource1">
                    <Columns>
                        <asp:BoundField DataField="sno"
HeaderText="sno" SortExpression="sno" />
                        <asp:BoundField DataField="name"
HeaderText="name" SortExpression="name" />
                        <asp:BoundField DataField="city"
HeaderText="city" SortExpression="city" />
                        <asp:BoundField DataField="class"
HeaderText="class" SortExpression="class" />
                    </Columns>
                </asp:GridView>
            </div>
        </form>
    </body>
</html>
```

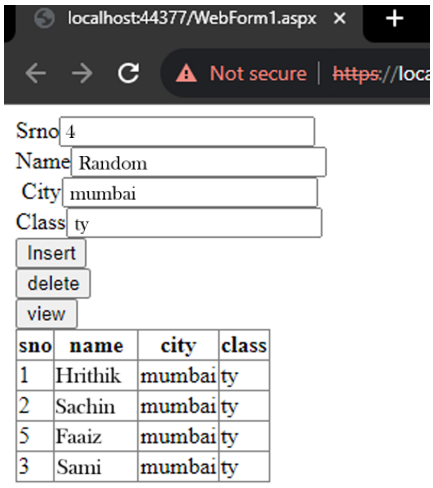
WebForm1.aspx.cs

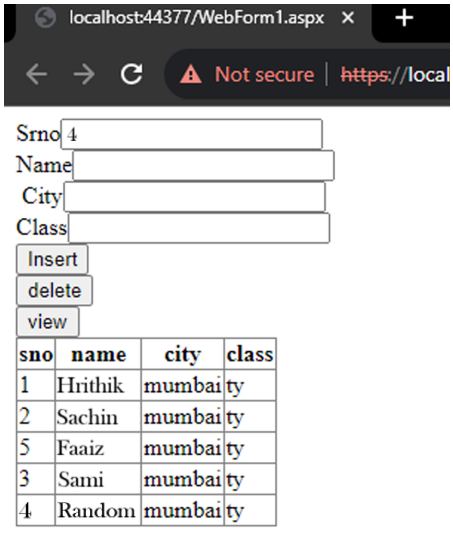
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
namespace Practical7_c {
    public partial class WebForm1: System.Web.UI.Page {
        SqlConnection cn = new SqlConnection("Data
Source=HP;Initial Catalog=db1;Integrated Security=True");
        SqlDataReader ds;
        SqlCommand co = new SqlCommand();
        SqlParameter @p1, @p2, @p3, @p4;
        protected void Button2_Click(object sender, EventArgs e)
{
            co.CommandText = "delete from student where sno= '"
+ TextBox1.Text + "';";
            co.ExecuteNonQuery();
        }
        protected void Button3_Click(object sender, EventArgs e)
{
            co.CommandText = "select * from student;";
            ds = co.ExecuteReader();
        }
        protected void Page_Load(object sender, EventArgs e) {
            cn.Open();
            co.Connection = cn;
        }
        protected void Button1_Click(object sender, EventArgs e)
{
            @p1 = new SqlParameter();
            @p1.ParameterName = "sno";
            @p1.SqlDbType = System.Data.SqlDbType.Int;
            @p2 = new SqlParameter();
            @p2.ParameterName = "name";
            @p2.SqlDbType = System.Data.SqlDbType.VarChar;
            @p3 = new SqlParameter();
            @p3.ParameterName = "city";
            @p3.SqlDbType = System.Data.SqlDbType.VarChar;
```

```
            @p4 = new SqlParameter();
            @p4.ParameterName = "class";
            @p4.SqlDbType = System.Data.SqlDbType.VarChar;
            co.Parameters.AddWithValue("@p1", TextBox1.Text);
            co.Parameters.AddWithValue("@p2", TextBox2.Text);
            co.Parameters.AddWithValue("@p3", TextBox3.Text);
            co.Parameters.AddWithValue("@p4", TextBox4.Text);
            co.CommandText = "insert into
student(sno,name,city,class) values(@p1,@p2,@p3,@p4);";
            co.ExecuteNonQuery();
        }
    }
}
```

Output:







10. Working with AJAX and XML.

10a Aim: Create a web application to demonstrate reading and writing operation with XML.

Source Code:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Practical10_A.WebForm1"
%>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head runat="server">
        <title></title>
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="Read Xml" />
                <br />
                <br />
                <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False">
                    <Columns>
                        <asp:BoundField DataField="name"
HeaderText="Name" />
                        <asp:BoundField DataField="roll"
HeaderText="Roll No" />
                    </Columns>
                </asp:GridView>
            </div>
        </form>
    </body>
</html>
```

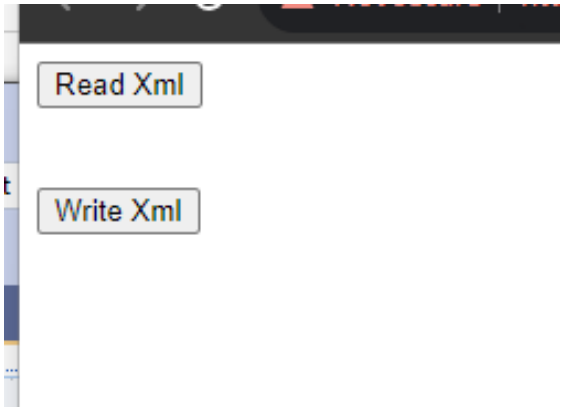
XMLFile1.xml:

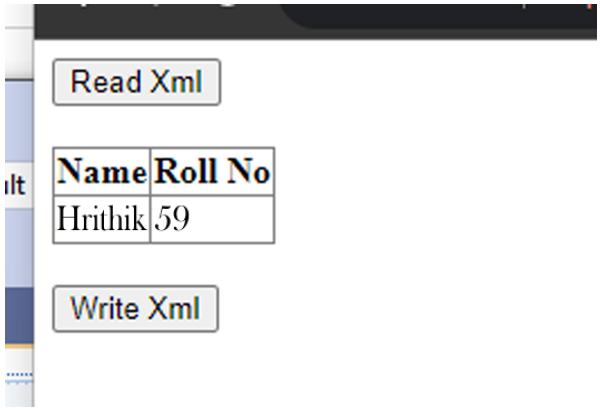
```
<?xml version="1.0" encoding="utf-8" ?>
     <student>
          <name> Aman </name>
          <roll> 58 </roll>
</student>
```

WebForm1.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Xml;
namespace Practical10_A {
    public partial class WebForm1: System.Web.UI.Page {
        protected void Page_Load(object sender, EventArgs e) {}
        protected void Button1_Click(object sender, EventArgs e)
{
            DataSet ds = new DataSet();
            ds.ReadXml(Server.MapPath(("XMLFile1.xml")));
            GridView1.DataSource = ds.Tables[0].DefaultView;
            GridView1.DataBind();
        }
        protected void Button2_Click(object sender, EventArgs e)
{
            XmlWriterSettings set = new XmlWriterSettings();
            set.Indent = true;
            using(XmlWriter xmlW =
                XmlWriter.Create(@
"C:\Users\puroh\source\repos\Practical10_A\Practical10_A\XmlFile
2.xml",
                    set)) {
                xmlW.WriteStartElement("Student");
                xmlW.WriteElementString("Name", "Aman");
                xmlW.WriteElementString("class", "Tyit");
                xmlW.WriteEndElement();
            }
        }}}
```

Output:






```
<?xml version="1.0 encoding="utf-8"?>
<Student>
    <Name> Hrithik </Name>
    <class> Tyit </class>
</Student>
```

10c Aim: Create a web application to demonstrate use of various Ajax controls.

Source Code:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="Pract_10c_ajaxcontrols.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
    <head runat="server">
        <title></title>
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                <asp:ScriptManager ID="ScriptManager1"
runat="server"></asp:ScriptManager>
                <br />
                <br />
```

```
                <asp:Button ID="Button1" runat="server"
OnClick="Button1_Click" Text="show time" />    
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
                <br />
                <asp:UpdateProgress ID="UpdateProgress1"
runat="server">
                    <ProgressTemplate> Please wait for some time
</ProgressTemplate>
                </asp:UpdateProgress>
                <br />
                <asp:UpdatePanel ID="UpdatePanel1"
runat="server"></asp:UpdatePanel>
                <br />
            </div>
        </form>
    </body>
</html>
```

WebForm1.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace Pract_10c_ajaxcontrols {
    public partial class WebForm1: System.Web.UI.Page {
        protected void Page_Load(object sender, EventArgs e) {
            System.Threading.Thread.Sleep(2000);
        }
        protected void Button1_Click(object sender, EventArgs e)
{
            TextBox1.Text = DateTime.Now.ToLongTimeString();
        }
    }
}
```

Output: